Reformulation semi-lisse appliquée au problème de complémentarité

par

Mathieu FRAPPIER

Mémoire présenté au Département de mathématiques
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, janvier 2019

Le 8 janvier 2019

le jury a accepté le mémoire de M. Mathieu Frappier dans sa version finale.

Membres du jury

M. Jean-Pierre Dussault
Directeur
Département d'informatique

M. Jean-Charles Gilbert
Codirecteur
Inria, Paris

M. Félix Camirand-Lemyre
Membre
Département de mathématiques

M. Tomasz Kaczynski
Président-rapporteur
Département de mathématiques

SOMMAIRE

Ce mémoire fait une revue des notions élémentaires concernant le problème de complémentarité. On y fait aussi un survol des principales méthodes connues pour le résoudre. Plus précisément, on s'intéresse à la méthode de Newton semi-lisse. Un article proposant une légère modification à cette méthode est présenté. Cette nouvelle méthode compétitive est démontrée convergente. Un second article traitant de la complexité itérative de la méthode de Harker et Pang est aussi introduit.

REMERCIEMENTS

Je voudrais d'abord remercier mon directeur de maîtrise Jean-Pierre Dussault pour ses précieux conseils et son financement. Sans son aide, il m'aurait été impossible de produire un tel document.

Je désire aussi remercier Jean Charles Gilbert de m'avoir transmis sa passion pour la recherche sur les problèmes de complémentarité. Je lui suis reconnaissant de m'avoir accueilli comme stagiaire et de m'avoir fait vivre une expérience des plus enrichissante à l'Inria de Paris. Une grande part du mérite lui revient aussi dans la réalisation de ce mémoire.

Ensuite, je tiens à remercier mes parents Richard Frappier et Linda Biron qui m'ont toujours soutenu dans mes études.

Merci au département de mathématiques de l'Université de Sherbrooke pour la formation que j'ai reçue ainsi que le financement fourni.

Finalement, je remercie mes collègues du laboratoire d'optimisation Luc, Maxime, Tangi, Émilie, Samuel, Jonathan et Paul pour leurs conseils et l'ambiance de travail toujours agréable.

Mathieu Frappier

Sherbrooke, août 2018

TABLE DES MATIÈRES

SOMMAIRE **i**

REMERCIEMENTS **ii**

TABLE DES MATIÈRES **iii**

INTRODUCTION **1**

CHAPITRE 1 — Revue de la littérature **3**

1.1 Préliminaires 3

1.2 Problèmes de complémentarité 6

1.3 Survol des principales méthodes connues 7

1.3.1 Newton-min 10

1.3.2 B-Newton 11

1.3.3 Harker et Pang 12

1.3.4 Han, Pang, Rangaraj 13

CHAPITRE 2 — Complexité itérative de l'algorithme de Harker et Pang 15

CHAPITRE 3 — Globalisation convergente de la méthode de Newton semi-lisse 36

CONCLUSION 62

BIBLIOGRAPHIE 68

INTRODUCTION

En toute généralité, le problème de complémentarité en dimension finie consiste à résoudre un système fini d'inéquations tout en respectant une équation particulière qui exprime la complémentarité entre les composantes. C'est cette caractéristique importante qui distingue le problème de complémentarité du système d'inéquations traditionnel.

L'intérêt d'étudier ce type de problème a commencé en 1964 lorsqu'il a été introduit par Richard W. Cottle dans sa thèse de doctorat puisque les applications sont nombreuses et dans plusieurs domaines différents. Tout d'abord, les problèmes de complémentarité sont apparus dans les conditions d'optimalité de Karush-Kuhn-Tucker [Kar39, KT51] mais peuvent aussi servir à modéliser certains phénomènes décrits par des systèmes d'équations qui sont en quelque sorte en compétition. Quelques exemples d'applications sont les problèmes d'équilibre économique [FP97], les jeux bimatriciels [Lem65, LJ64], le problème d'équilibre du trafic de Wardrop [War52], les problèmes d'écoulement diphasiques [BJ14, BKKK11, BGS13, BE18] et les simulations de contacts et de mouvements de fluides [Erl13]. Une des raisons qui explique pourquoi les problèmes de complémentarité modélisent bien ces applications est que le concept de complémentarité est intimement relié à la notion d'équilibre. Au fil des années, le sujet est devenu une discipline proprement dite des mathématiques. Les contributions à la littérature concernant les problèmes de complémentarité ont été apportées par des mathématiciens, mais aussi par des infor-

maticiens, des économistes et des ingénieurs dû à leur vaste champ d'application.

Ce mémoire est divisé en trois chapitres. D'abord, dans le premier chapitre, des définitions importantes et quelques résultats intéressants sont donnés. Ensuite, un état de l'art est présenté.

Le second chapitre traite de la complexité itérative de l'algorithme de Harker et Pang [HP90]. Un article soumis qui donne une borne inférieure sur le nombre d'itérations requises en pire cas pour résoudre un problème de complémentarité linéaire est présenté.

Finalement, le troisième chapitre présente un article en cours de finition où un algorithme inspiré de la méthode de Newton semi-lisse est proposé. Cet algorithme possède la propriété de convergence globale et contrairement aux autres algorithmes similaires, il ne résout pas de sous-problèmes de complémentarité linéaire.

Bien que les notions de base concernant les problèmes de complémentarité soient reprises depuis le début, on suppose du lecteur qu'il possède déjà une solide connaissance en analyse mathématique.

CHAPITRE 1

Revue de la littérature

Avant d'aborder le sujet des problèmes de complémentarité, nous rappelons d'abord quelques définitions et propriétés qui seront utilisées au cours du mémoire.

1.1 Préliminaires

À moins d'indication contraire, dans cette section nous considérons le cas où f est une fonction de $\mathbb{R}^n \to \mathbb{R}^n$.

Définition 1.1. *Une fonction f est dite de Lipschitz ou lipschitzienne s'il existe une constante L telle que pour tout $x, y \in \mathbb{R}^n$ la relation*

$$\|f(x) - f(y)\| \leq L\|x - y\|$$

est vérifiée. La constante L est appelée constante de Lipschitz.

Définition 1.2. *Une fonction f est dite localement de Lipschitz ou localement lipschitzienne en un point $x \in \mathbb{R}^n$ s'il existe un entier positif ε tel que f est de Lipschitz sur la boule $B(x, \varepsilon)$.*

Dans ce mémoire, il sera question de fonctions qui ne sont pas différentiables donc nous devons généraliser le concept de la dérivée. Nous allons introduire deux sous-différentiels qui sont reliés entre eux. L'ensemble des points où f est différentiable sera noté D_f et le jacobien de f en x sera noté $\nabla f(x)$.

Définition 1.3. *Le B-sous-différentiel (pour Bouligand) de f évalué en x est défini par*

$$\partial_B f(x) = \{J : (x_k) \subset D_f, x_k \to x, \nabla f(x_k) \to J\}.$$

De manière générale, ce différentiel est difficile à calculer et à manipuler. Il ne permet pas d'avoir des conditions d'optimalité en optimisation ni de théorème de la moyenne. On obtient de meilleures propriétés en prenant son enveloppe convexe ce qui conduit au Jacobien généralisé de Clarke.

Définition 1.4. *Le Jacobien généralisé de Clarke [Cla83] de f en x noté $\partial f(x)$ est l'enveloppe convexe de $\partial_B f(x)$.*

Définition 1.5. *Étant donné une fonction $f : \mathbb{R}^n \to \mathbb{R}^m$ définie par*

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_m(x) \end{bmatrix}.$$

Le sous-différentiel de Clarke de f en x est défini par

$$\partial_C f(x) = \partial f_1(x) \times \partial f_2(x) \times \cdots \times \partial f_m(x)$$

où $\partial f_i(x)$ dénote le Jacobien généralisé de Clarke de $f_i(x)$.

Ce qu'on veut dire par cette définition dans [Cla83] est que $\partial_C f(x)$ est l'ensemble des matrices dont chaque ligne i est un élément de $\partial f_i(x)$. On remarque que lorsque $m = 1$, il s'ensuit que $\partial_C f(x) = \partial f(x)$.

Notons que si f est localement lipschitzienne et continue, par le théorème de Rademacher l'ensemble des points où f est non-différentiable a une mesure de Lebesgue nulle. Cela entraîne que f est différentiable presque partout et le résultat suivant.

Proposition 1.6. *Soit $f : \mathbb{R}^n \to \mathbb{R}^n$ localement lipschitzienne et continue. Alors*

1. *$\partial_B f(x) \subseteq \partial f(x) \subseteq \partial_C f(x)$,*
2. *$\partial f(x)$ est un ensemble compact, convexe et non-vide,*
3. *f est continûment différentiable si et seulement si $\partial f(x) = \{f'(x)\}$.*

Démonstration. Voir la proposition 2.6.2 et la proposition 2.2.4 de [Cla83]. □

Dans ce qui suit, nous aurons aussi besoin de la notion de fonction semi-lisse. L'ensemble de ces fonctions est un sous-ensemble des fonctions localement lipschitziennes. D'abord, rappelons la notion de dérivée directionnelle.

Définition 1.7. *Une fonction f est directionnellement différentiable en x si la limite*

$$f'(x; d) = \lim_{t \to 0} \frac{f(x + td) - f(x)}{t}$$

existe pour toutes les directions $d \in \mathbb{R}^n$.

Définition 1.8. *Soit f localement lipschitzienne et directionnellement différentiable en x. On dit que f est semi-lisse en x si*

$$\lim_{d \to 0, G \in \partial f(x+d)} \frac{Gd - f'(x; d)}{\|d\|} = 0.$$

De plus, f est fortement semi-lisse en x si

$$\lim_{d \to 0, G \in \partial f(x+d)} \frac{Gd - f'(x; d)}{\|d\|^2} < \infty.$$

Une dernière définition sera utile à la définition du problème de complémentarité. Il s'agit de perpendicularité.

Définition 1.9. *Soient $x \in \mathbb{R}^n$ et $y \in \mathbb{R}^n$, on dit que x est perpendiculaire à y si $x^\mathsf{T} y = 0$ et on le note $x \perp y$.*

1.2 Problèmes de complémentarité

Soit le problème de complémentarité suivant sous sa forme réduite

$$0 \leq x \perp f(x) \geq 0 \tag{CP}$$

où $x \in \mathbb{R}^n$ et $f : \mathbb{R}^n \to \mathbb{R}^n$ (CP pour *Complementarity Problem*). D'abord, notons que lorsque $n \geq 2$, les inégalités doivent être prises composante par composante. Ainsi, le problème de complémentarité (CP) est de trouver un vecteur x positif qui satisfait $f(x) \geq 0$, mais aussi l'équation de complémentarité qui veut que pour une composante i donnée soit x_i s'annule, soit $f_i(x)$ s'annule. Puisque les deux sont positifs, l'équation de complémentarité peut s'exprimer comme $x \perp f(x)$.

Une classe de problèmes de complémentarité intéressante à étudier est celle des problèmes linéaires. C'est-à-dire lorsque $f(x) = Mx + q$ avec $M \in \mathbb{R}^{n \times n}$. On notera ce problème LCP(M, q) (pour *Linear Complementarity Problem*). Nous présentons ici certaines classes de matrices qui ont des propriétés importantes pour les problèmes de complémentarité linéaires.

Définition 1.10. *Une matrice carrée M est une $\boldsymbol{P}$-matrice si tous ses mineurs principaux sont strictement positifs. Similairement, une matrice carrée M est une $\boldsymbol{P}_0$-matrice si tous ses mineurs principaux sont positifs.*

Notons que Fiedler et Pták[FP62] ont démontré en 1962 que $M \in \mathbf{P}$ si et seulement si toutes les valeurs propres réelles de M et de ses sous-matrices carrées principales sont strictement positives. Ainsi, toute matrice symétrique définie positive est une **P**-matrice. L'importance de ces matrices vient du résultat d'existence et d'unicité suivant.

Théorème 1.11 (Théorème 3.3.7 [CPS92])**.** *Une matrice $M \in \mathbb{R}^{n \times n}$ est une $\boldsymbol{P}$-matrice si et seulement si le problème LCP(M, q) admet une unique solution pour tout $q \in \mathbb{R}^n$.*

Définition 1.12. *Une matrice carrée M, pour laquelle il existe un vecteur x tel que*

$$Mx > 0, \quad x > 0$$

est appelée une ***S****-matrice.*

L'intérêt pour cette classe de matrice vient du fait que l'on peut démontrer que $M \in \mathbf{S}$ si et seulement si $\mathrm{LCP}(M, q)$ est réalisable pour tout vecteur $q \in \mathbb{R}^n$ [CPS92, Proposition 3.1.5].

Remarque 1.13. *Dans la littérature, on dit que* (CP) *admet une solution (solvable) s'il existe un vecteur x qui satisfait aux contraintes de positivité et à la complémentarité. Alors qu'on dit qu'il est réalisable (feasible) s'il existe $x \geq 0$ tel que $f(x) \geq 0$.*

On se permet de faire cette distinction puisque souvent c'est l'équation de complémentarité qui est la plus difficile à respecter.

Tout au long du mémoire, pour une matrice $M \in \mathbb{R}^{n \times n}$ et les ensembles d'indices I et $J \subseteq [1:n]$ nous désignerons par M_{IJ} la sous-matrice de M formée des lignes des éléments de I et des colonnes formées des éléments de J.

1.3 Survol des principales méthodes connues

Depuis l'apparition des problèmes de complémentarité, plusieurs méthodes ont été mises au point afin d'en calculer une solution. La première méthode d'importance à faire son apparition est l'algorithme de Lemke [Lem65]. Il s'agit d'une méthode s'apparentant à celle du Simplexe en optimisation linéaire qui calcule une solution de base réalisable et par opérations de pivots détermine le prochain itéré. Cette méthode est seulement définie pour des problèmes linéaires et dans certains cas peut prendre un nombre d'itérations

très élevé. Il a été montré que cette méthode a une complexité itérative exponentielle [Mur78].

Une autre approche qui a été étudiée est la méthode de points intérieurs. Telle que décrite par Facchinei et Pang [FP03, chapitre 11], cette classe d'algorithme suppose que l'on connaît un point initial dans l'ensemble des points strictement réalisables

$$\Gamma = \{x : x > 0, f(x) > 0\}.$$

Par la suite, on s'approche d'un point complémentaire en pénalisant de plus en plus le résidu sur la contrainte de complémentarité tout en conservant $x \in \Gamma$ au fil des itérations. Plus tard, Wright et Zhang [Wri94, Zha94] ont proposé de conserver uniquement $x > 0$ afin que les itérations soient moins coûteuses. Cette variante est appelée méthode de points intérieurs non-réalisables (infeasible interior-points method).

Les méthodes de projections dont le "path search" [DF95] ont aussi beaucoup été analysées. Dans le cas non-linéaire, le "path search" consiste à linéariser f autour de la projection du point courant x sur l'orthant positif. Ensuite, un chemin linéaire par morceaux de x à $x_N(x)$ le point de Newton est produit en résolvant un LCP par une méthode de pivot s'apparentant à celle de Lemke. Une itération se termine par une recherche visant à réduire une certaine fonction de mérite sur ce chemin.

Une autre façon d'aborder le problème est en le reformulant à l'aide d'autres équations. Une reformulation importante est celle de Fischer-Burmeister. En posant $\phi(a, b) = \sqrt{a^2 + b^2} - a - b$, la méthode consiste à trouver un zéro de la fonction $\phi(x, f(x))$. Il est évident que si x satisfait ces conditions, ce vecteur est aussi solution de (CP). Initialement l'algorithme de Fischer [Fis92] utilisait la direction de Newton pour trouver un zéro de $\phi(x, f(x))$, mais d'autres variantes ont été apportées par la suite dont la pénalisation de B.Chen X.Chen et Kanzow, qui selon eux améliore significativement la performance de l'algorithme [CCK00]. Deux inconvénients importants de cette reformu-

lation sont qu'elle n'assure pas la terminaison finie de l'algorithme pour les problèmes linéaires et qu'elle n'assure pas que tous les éléments du B-différentiel en la solution sont inversibles [LFK00]. Ce manque de régularité en la solution peut faire en sorte que la convergence locale asymptotique de cette méthode soit plus lente [LFK00] alors que la prochaine reformulation avec la fonction min ne présente pas ces défauts.

Finalement, la méthode qui nous intéresse et dont il sera question dans les prochains chapitres est la reformulation du problème en un système d'équations non-lisse. Une attention particulière sera accordée à la méthode de Newton semi-lisse. Il est facile de démontrer que (CP) est équivalent à trouver un zéro à la fonction $F(x) = \min(x, f(x))$, le minimum composante par composante où $F : \mathbb{R}^n \to \mathbb{R}^n$. À cause de la fonction minimum, F n'est pas différentiable. Par contre, si f est différentiable, ce qui sera toujours supposé dans ce qui suit alors F est semi-lisse. Notons que les points de non-différentiabilité sont seulement ceux pour lesquels $x_i = (f(x))_i$ pour au moins un indice i. Ces points sont appelés des *plis* de F. Nous verrons que ce sont ces points qui posent problème dans la plupart des algorithmes de type Newton semi-lisse. En toute généralité, puisque F n'est pas différentiable, les algorithmes de type Newton semi-lisse utilisent J un élément inversible du sous-différentiel de Clarke et calculent une direction d qui satisfait l'équation de Newton

$$F(x) + Jd = 0.$$

Le prochain itéré est donc $x + d$. Lorsqu'elle converge, cette méthode possède la propriété de convergence locale quadratique typique à la méthode de Newton.

Nous présentons ici quatre algorithmes de type Newton semi-lisse importants et qui ont inspiré l'algorithme Newton-min-hybrid qui sera présenté au chapitre 3.

1.3.1 Newton-min

Une méthode de type Newton semi-lisse très simple et souvent efficace en pratique est l'algorithme Newton-min. D'abord, définissons les ensembles suivants

$$\begin{array}{rcl} A_0 \equiv A_0(x) & := & \{i : x_i < f_i(x)\}, \\ I_0 \equiv I_0(x) & := & \{i : x_i > f_i(x)\}, \\ E \equiv E(x) & := & \{i : x_i = f_i(x)\}. \end{array} \tag{1.1}$$

L'algorithme Newton-min choisit une partition E_A et E_I de E telle que $E_A \cup E_I = E$ et $E_A \cap E_I = \varnothing$. Ensuite, on pose $A = A_0 \cup E_A$ et $I = I_0 \cup E_I$ où les indices dans A seront ceux à activer et ceux en I seront inactifs en x. En effet, si le point courant n'est pas trop loin d'une solution, il est naturel de choisir le prochain itéré x^+ tel que $x_A^+ = 0$ et $f_I(x^+) = 0$. Puisque la seconde égalité peut être plus difficile à obtenir, on calcule le zéro de l'approximation linéaire de f. Ainsi, supposant que toutes les sous-matrices principales de $\nabla f(x)$ sont inversibles, x^+ est l'unique solution du système d'équations

$$x_A^+ = 0, \quad (f(x) + \nabla f(x)(x^+ - x))_I = 0. \tag{1.2}$$

On remarque que la direction de Newton-min n'est pas toujours unique, car le système d'équations (1.2) dépend du choix de partition de E. Par contre lorsque $E(x)$ est vide, x est un point où f est différentiable donc la direction est unique. Remarquons qu'en posant $d^{\text{NM}} = x^+ - x$, après quelques manipulations simples on obtient que la direction de Newton-min est

$$d_A^{\text{NM}} = -x_A, \quad d_I^{\text{NM}} = -\nabla f_{II}(x)^{-1}(f_I(x) + \nabla f_{IA}(x) d_A).$$

Par conséquent, on remarque qu'il peut être plus efficace de toujours choisir $E_A = E$ et $E_I = \varnothing$ afin de réduire la taille du système d'équations à résoudre en I.

Malheureusement, cette méthode n'est pas convergente. En fait, on peut trouver des exemples où l'algorithme cycle [BG13, BG18]. La difficulté à globaliser cette méthode

vient du fait que la direction d n'est pas toujours une direction de descente de la fonction de mérite des moindres carrés définie par

$$\Theta(x) = \tfrac{1}{2}\|F(x)\|^2.$$

Par conséquent, à ce jour aucune preuve montrant que Newton-min avec recherche linéaire est convergent n'est connue. D'ailleurs, un algorithme utilisant cette stratégie a été proposé dans [BD06] et les auteurs prétendent avoir la propriété de convergence globale. Or une erreur a été faite dans la démonstration et un contre-exemple montrant que les pas de déplacement peuvent tendre vers zéro est présenté en annexe ce qui détruit un argument utilisé dans leur tentative de preuve de convergence. Malgré cela, nous verrons au chapitre 3 que Newton-min est très efficace en pratique. Le faible coût de calcul de d^{NM} en fait une direction intéressante. D'ailleurs celle-ci est utilisée dans certains algorithmes dont [HP90, DFG18, BD06].

1.3.2 B-Newton

Parmi les $2^{\mathbf{card}(E)}$ manières différentes de partitionner E dans l'algorithme de Newton-min, il a été montré dans [Pan90] qu'au moins un choix mènera à une direction de descente de Θ. Étant donné que la fonction de mérite à minimiser Θ n'est pas différentiable partout, Pang propose d'utiliser la B-dérivée (Bouligand) dans la méthode de Newton. Ainsi, la direction de B-Newton d^{B}, en un point x, est donnée par la solution de l'équation

$$F(x) + BF(x)d^{\text{B}} = 0 \tag{1.3}$$

où $BF(x)$ est un élément du B-sous-différentiel de F évaluée en x. En développant (1.3), on obtient que d^{B} est l'unique solution du problème de complémentarité mixte suivant

$$\begin{cases} (x + d^{\text{B}})_{A_0} = 0 \\ (f(x) + \nabla f(x)d^{\text{B}})_{I_0} = 0 \\ 0 \leq (x + d^{\text{B}})_E \perp (Mx + q + Md^{\text{B}})_E \geq 0. \end{cases} \tag{1.4}$$

Un inconvénient de cette méthode est qu'elle doit résoudre des sous-problèmes de complémentarité linéaire ce qui peut ne pas faire de sens lorsque le problème initial (CP) est linéaire. Un exemple où le problème initial et le calcul de la direction (1.4) est exactement le même problème est présenté dans [BG12]. De plus, la preuve de convergence globale de l'algorithme donnée par Pang suppose que les pas de déplacement sont uniformément positifs ce qui est une hypothèse très forte.

1.3.3 Harker et Pang

Une autre manière d'aborder le problème est la méthode de Harker et Pang [HP90]. Leur méthode a été présentée pour résoudre des problèmes de complémentarité linéaire où toutes les sous-matrices principales de $M = \nabla f(x)$ sont inversibles. Puisque les points de non-différentiabilité sont ceux qui posent problème, ces auteurs proposent une variante de la méthode de Newton-min ou B-Newton qui les évite à tout prix.

D'abord, supposons que $E(x) = \varnothing$. Dans ce cas, les équations de Newton-min (1.2) et de B-Newton (1.4) sont les mêmes. Nous savons que cette direction est unique mais en plus, il est facile de démontrer que la dérivée directionnelle $\Theta'(x; d^{\text{NM}}) = -2\Theta(x)$. Par conséquent, d^{NM} est une direction de descente pour Θ tant que x n'est pas la solution. C'est pourquoi avec un point de départ tel que $E(x) = \varnothing$, le fonctionnement de l'algorithme est d'abord de vérifier si $x + d^{\text{NM}}$ est une solution de (CP) et sinon de calculer $\breve{\alpha}_1 \in (0, 1)$ le plus petit pas de déplacement tel que $E(x + \breve{\alpha}_1 d^{\text{NM}}) \neq \varnothing$. Par la suite, l'idée est de prendre comme prochain itéré un point pas trop loin de l'autre côté du premier pli rencontré le long de d^{NM} parce que f est linéaire et donc $\Theta(x)$ est quadratique par morceaux et décroissant le long de d^{NM} jusqu'à $x + \breve{\alpha}_1 d^{\text{NM}}$. Par conséquent, le prochain itéré est $x + (\breve{\alpha}_1 + \varepsilon)d$ pour une petite valeur de ε. Si $x + (\breve{\alpha}_1 + \varepsilon)d$ ne tombe jamais sur un pli, alors le prochain itéré est toujours un point différentiable et facilement calculable comme pour Newton-min. En

choisissant un pas de déplacement supérieur à $\breve{\alpha}_1$, nous nous assurons que les ensembles d'indices A_0 et I_0 changent d'une itération à l'autre changeant aussi la direction calculée. Par contre rien n'assure que $x + (\breve{\alpha}_1 + \varepsilon)d^{\text{NM}}$ n'est pas sur un pli. De plus, rien ne garantit que $\Theta(x + \alpha d)$ continue de décroître après le premier pli, ce pourquoi il serait risqué de prendre ε trop grand. Afin de surmonter ces deux problèmes, les auteurs suggèrent d'utiliser le processus d'Armijo tout en assurant que le pas de déplacement soit supérieur à $\breve{\alpha}_1$. Une technique similaire afin d'éviter ce danger sera également présentée au chapitre 2.

Un problème avec cette méthode est qu'en arithmétique flottante l'ensemble E n'est pas bien défini. De plus, dans certains problèmes de grande taille, il serait possible de trouver plusieurs plis de F amassés le long de $x + d^{\text{NM}}$. Dans cette situation, il peut être très difficile de trouver une bonne valeur ε, voire impossible à cause de l'arithmétique flottante finie. À l'époque où l'algorithme a été présenté, les problèmes de grande dimension ne pouvaient pas être traités et on constate aujourd'hui qu'il est difficilement applicable pour ces derniers.

1.3.4 Han, Pang, Rangaraj

Au meilleur de ma connaissance, la seule globalisation connue de la méthode de Newton semi-lisse est celle de Pang [Pan91] qui a été reprise par Han, Pang et Rangaraj dans [HPR92]. Comme pour la méthode de B-Newton, ils proposent de calculer un élément du B-sous-différentiel afin d'obtenir une direction de descente de Θ. Cependant, ils considèrent des ensembles d'indices différents de (1.1). Ils remarquent que pour la convergence de leur algorithme, il est nécessaire d'ajouter au LCP les indices pour lesquels x_i et $f_i(x)$

sont tous deux négatifs. Donc si on définit les ensembles

$$\begin{aligned} A_{HPR} \equiv A_{HPR}(x) &:= \{i : x_i < f_i(x), f_i(x) \geq 0\}, \\ I_{HPR} \equiv I_{HPR}(x) &:= \{i : x_i > f_i(x), x_i \geq 0\}, \\ E_{HPR} \equiv E_{HPR}(x) &:= \{i : [1:n] \backslash (A_{HPR} \cup I_{HPR})\}, \end{aligned}$$

la direction de Han, Pang et Rangaraj se calcule de la même manière que celle de B-Newton mais avec ceux-ci. Ensuite, une itération se termine avec une recherche linéaire de long de cette direction.

Un point fort de cet algorithme est qu'il est globalement convergent. En contrepartie, de manière générale les itérations sont fort coûteuses à cause de la taille du LCP qui peut être très grande. La définition de cette méthode est reprise dans l'article présenté au chapitre 3 car elle a inspiré notre variante de la méthode de Newton.

CHAPITRE 2

Complexité itérative de l'algorithme de Harker et Pang

Résumé

Le but de cet article est de présenter une borne inférieure sur le nombre d'itérations requises par l'algorithme de Harker et Pang en pire cas pour résoudre un problème de complémentarité linéaire. Pour y arriver, on propose d'abord une légère modification de l'algorithme original de Harker et Pang afin de s'assurer de ne pas aller trop loin dans la direction calculée lorsqu'il y a plusieurs plis le long de d. On propose de calculer les deux plus petits pas de déplacement $\check{\alpha}_1$ et $\check{\alpha}_2$ tels que $x + \check{\alpha}_i d$ est sur un pli de Θ et de choisir un pas de déplacement dans l'intervalle $(\check{\alpha}_1, \check{\alpha}_2)$. Ensuite, une étude rigoureuse du problème de Fathi [Fat79] est réalisée. On y montre que la matrice qui définit le problème possède deux caractéristiques importantes qui permettront de conclure que pour un point initial dans un voisinage de zéro, l'algorithme modifié prend exactement n itérations pour trouver la solution où n est le nombre de variables. Finalement, on observe qu'il existera toujours une valeur $\varepsilon > 0$ telle que l'algorithme de Harker et Pang prend aussi n itérations

pour trouver la solution lorsque le point de départ est dans un voisinage de l'origine. Nous obtenons ainsi n comme borne minimale sur la complexité itérative.

Commentaires

Pour commencer, ma contribution à cet article a été l'implémentation de l'algorithme de Harker et Pang en Matlab afin de reproduire leurs résultats. Après avoir testé le problème de Fathi avec plusieurs points de départ et en plusieurs dimensions, j'ai constaté que l'algorithme ne prenait jamais plus de n itérations pour trouver la solution. Toutefois, ce qui était le plus surprenant c'est qu'il semblait toujours prendre le même "chemin". Après quelques itérations seulement, la suite des ensembles d'indices $A_0(x_i)$ et $I_0(x_i)$ était identique pour tous les problèmes jusqu'à la solution. C'est ce qui m'a fait conjecturer que pour toute valeur n, il était possible de trouver un voisinage de l'origine pour lequel l'algorithme prendrait exactement n itération pour trouver la solution avec la bonne valeur ε. Finalement, j'ai participé à l'élaboration de la preuve de cette conjecture. Cet article a été soumis à EURO Journal on Computational Optimization.

A lower bound on the iterative complexity of the Harker and Pang globalization technique of the Newton-min algorithm for solving the linear complementarity problem

J.-P. Dussault [†], M. Frappier [‡], and J. Ch. Gilbert [§]

Tuesday 11th December, 2018

The plain Newton-min algorithm for solving the linear complementarity problem (LCP) "$0 \leqslant x \perp (Mx+q) \geqslant 0$" can be viewed as an instance of the plain semismooth Newton method on the equational version "$\min(x, Mx+q) = 0$" of the problem. This algorithm converges, whatever is q, when M is an **M**-matrix, but not when it is a **P**-matrix. When convergence occurs, it is often very fast (in at most n iterations for an **M**-matrix, where n is the number of variables, but often much faster in practice). In 1990, Harker and Pang proposed to improve the convergence ability of this algorithm by introducing a stepsize along the Newton-min direction that results in a jump over at least the first encountered kink of the min-function, in order to avoid its points of nondifferentiability. This paper shows that, for the Fathi problem (an LCP with a positive definite symmetric matrix M, hence a **P**-matrix), an algorithmic scheme, including the algorithm of Harker and Pang, may require n iterations to converge, depending on the starting point.

Keywords: iterative complexity, linear complementarity problem, Fathi and Murty problems, globalization, Harker and Pang algorithm, linesearch, Newton-min algorithm, nondegenerate matrix, **P**-matrix, semismooth Newton method.

AMS MSC 2010: 15B99, 47B99, 49M15, 65K15, 90C33.

1 Introduction

Let $n \geqslant 1$ be an integer, $M \in \mathbb{R}^{n\times n}$ be a real matrix, $q \in \mathbb{R}^n$ be a real vector, and $[1\,{:}\,n] := \{1, \ldots, n\}$ be the set of the first n positive integers. The linear complementarity problem (LCP) consists in searching a vector $x \in \mathbb{R}^n$ such that

$$0 \leqslant x \perp (Mx+q) \geqslant 0. \tag{1.1}$$

This means that the sought x must satisfy $x \geqslant 0$, $Mx+q \geqslant 0$ (vectorial inequalities must be understood componentwise), and $x^{\mathsf{T}}(Mx+q) = 0$ (the exponent "$^{\mathsf{T}}$" is used to denote matrix transposition). The problem has a combinatorial aspect, which lies in this last equation, since, by the nonnegativity of x and $Mx+q$, it amounts to the set of n complementarity conditions $x_i(Mx+q)_i = 0$ for all indices $i \in [1\,{:}\,n]$. The term complementarity comes from the fact that, for all $i \in [1\,{:}\,n]$, either x_i or $(Mx+q)_i$

[†]Département d'Informatique, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: Jean-Pierre.Dussault@Usherbrooke.ca.

[‡]Département de Mathématiques, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: Mathieu.Frappier@usherbrooke.ca.

[§]INRIA Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France. E-mail: Jean-Charles.Gilbert@inria.fr.

must vanish; conditions that may be realized in 2^n different ways. Actually, the problem of determining whether a particular instance of the LCP has a solution is strongly NP-complete [14], and NP-complete for a $\mathbf{P_0}$-matrix (i.e., when M has nonegative principal minors) [28].

Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be the *min-function* associated with the LCP (1.1), which is the function that takes at $x \in \mathbb{R}^n$ the value

$$F(x) = \min(x, Mx + q). \tag{1.2}$$

The Newton-min algorithm can be viewed as an instance of the semismooth Newton method [39] to solve the equational equivalent form of (1.1) [31, 32, 16] that reads $F(x) = 0$. To write compactly the algorithm, it is useful to introduce, for $I \subseteq [1\,{:}\,n]$ and its complement $A := [1\,{:}\,n] \setminus I$, the point $x^{(I)}$ defined by

$$x_A^{(I)} = 0 \qquad \text{and} \qquad (Mx^{(I)} + q)_I = 0,$$

This point is well defined when M is *nondegenerate*, meaning that its principal minors do not vanish. The plain *Newton-min algorithm* computes the next iterate by

$$\hat{x} := x^{(\mathcal{S}(x))}, \tag{1.3}$$

where the *index selector* $\mathcal{S} : \mathbb{R}^n \multimap [1\,{:}\,n]$ is the multifunction defined at $x \in \mathbb{R}^n$ by

$$\mathcal{S}(x) := \{i \in [1\,{:}\,n] : x_i > (Mx + q)_i\}. \tag{1.4}$$

In some versions of the algorithm, $\mathcal{S}(x)$ also contains some or all the indices in $\{i \in [1\,{:}\,n] : x_i = (Mx + q)_i\}$. See paragraph 7 of the introduction of [4] for more details on the origin of this algorithm and a discussion on the contributions from [12, 30, 22, 21, 8, 7, 23, 27]. When the current iterate $x \in \mathbb{R}^n$ is not on a kink of F, like in this paper, the Newton-min algorithm is identical to the Newton method to find a zero of F, which is then well defined.

Even though the Newton-min algorithm uses no globalization technique, like linesearches or trust regions [9, 15], it may converge globally, i.e., from any starting point. This is due to the very particular piecewise linearity of F. For example, global convergence occurs, whatever is q, when M is an $\mathbf{M}$-matrix [1], which is a $\mathbf{P}$-matrix (i.e., with positive principal minors) with nonpositive off-diagonal elements. It also occurs when M is close enough to an $\mathbf{M}$-matrix [23]. However, this global convergence property does not extend up to the larger class of $\mathbf{P}$-matrices [4, 5, 17, 6]. This is unfortunate, since $\mathbf{P}$-matrices are exactly those ensuring the existence and uniqueness of the solution to the LCP, whatever is q [40, 16].

A natural idea to enlarge the class of matrices, for which the global convergence of the Newton-min algorithm can be guaranteed, is to introduce linesearch on the associated *least-square merit function*, which is the function $\Theta : \mathbb{R}^n \to \mathbb{R}$ defined at $x \in \mathbb{R}^n$ by

$$\Theta(x) = \frac{1}{2}\|F(x)\|^2,$$

where $\|\cdot\|$ denotes the Euclidean norm. This least-square function is natural, since it has been used, often with success, for globalizing the Newton method when the function F is smooth [18, 15, 9, 25]. In the presence of nonsmoothness of F, like here, this technique

is more difficult to implement, since the Newton-min direction $d := \hat{x} - x$ may not be a descent direction of Θ at a kink of F [3]. To overcome this difficulty, Harker and Pang proposed in [22; p. 275] a method named the *Modified Damped-Newton Algorithm*, which consists in taking for the next iterate the point

$$x^+ := x + (\check{\alpha}_1 + \varepsilon_{\text{HP}})d,$$

where $\check{\alpha}_1 > 0$ is a stepsize so that $x + \check{\alpha}_1 d$ is on the first kink of F encountered along d from x, and $\varepsilon_{\text{HP}} > 0$ is a number such that the new iterate x^+ is not on a kink of F and ensures a sufficient decrease of the least-square merit function Θ. Consequently, this algorithm avoids the points of nondifferentiability of F, generates descent directions of Θ, and forces Θ to decrease sufficiently at each iteration. To the best of our knowledge, the only convergence result for any linesearch algorithm using a semi-smooth Newton direction uses the assumption that $\liminf_k \alpha_k > 0$ [37], which is a very weak result since this strong assumption relates to the algorithm products rather than the problem data.

In this research field, sparing of theoretical results, this paper provides the value n as a worse case lower bound on the number of iterations of the Harker and Pang algorithm when the extra stepsize ε_{HP} is taken sufficiently small, which is allowed by the description of the method given in [22; p. 275]. This lower bound is obtained on the Fathi problem for a set of starting points, including the one of [20], which is zero. To extend the applicability of this result, we describe an algorithmic scheme, for which this worse case lower bound is valid; a scheme that includes the Harker and Pang algorithm for sufficiently small positive ε_{HP}. In this scheme, the iterates avoid the kinks of F and the stepsizes are chosen arbitrarily between the first two break-stepsizes $\check{\alpha}_1$ and $\check{\alpha}_2$ (to be defined). Now, on many practical problems, an algorithm using the Newton-min direction and a stepsize that is not forced to be in $(\check{\alpha}_1, \check{\alpha}_2)$ usually finds a solution in much less iterations than n; in the experiments of [3], it is not uncommon to encounter LCPs having up to 10^5 variables that are solved in less than 10 iterations. Nevertheless, the Fathi problem remains a difficult instance of LCP for this family of methods, independently of the chosen stepsizes. To illustrate this, we show in the numerical experiment section that, surprisingly, doing exact linesearches hardly modifies the iteration counter. Finally, this worse case lower bound and the numerical experiments of section 5 suggest that it is unlikely that the improvement of the Newton-min algorithm can lie in a better determination of the stepsizes. This observation paves the way for the proposals made in [3].

To conclude this introduction, let us mention that there are a large number of contributions related to the complexity of algorithms for solving the LCP. Most of them are related to interior point methods and it is out of the scope of this paper to review them (they can be found by looking at those citing one of the first accounts on the subject, which is [28, 29]). Other approaches are sometimes qualified as noninterior path-following/continuation methods and are based on the smoothing of equational versions of the LCP: the function $(a, b) \in \mathbb{R}^2 \mapsto a + b - [(a - b)^2 + 4\mu^2]^{1/2}$ is considered in [13, 26] and the smooth Fisher-Burmeister function $(a, b) \in \mathbb{R}^2 \mapsto a + b - [a^2 + b^2 + 2\mu^2]^{1/2}$ is used in [26]. The complexity of these approaches have been studied in [10, 24, 11], for instance.

The paper is structured as follows. The algorithmic scheme, for which the lower bound on the iterative complexity is obtained, is presented in section 2. The Fathi problem and two properties of its matrix are given in section 3. The iterative complexity result is proved

in section 4. Finally some numerical experiments are reported in section 5 and the paper ends with the conclusion section 6.

Notation. For the $n \times n$ matrix M and index sets I and $J \subseteq [1\,{:}\,n]$, we denote by M_{IJ} the submatrix of M formed of its elements with row indices in I and column indices in J. We also define $M_{I:} := M_{I[1\,:\,n]}$ and $M_{II}^{-1} := (M_{II})^{-1}$.

2 The Newton-min-HP-ext algorithmic scheme

In [22; 1990, p. 275], Harker and Pang proposed a method to solve the LCP (1.1) that they named the *Modified Damped-Newton Algorithm*. It is grounded on Newton's iterations to find a zero of the function F defined in (1.2), and it is first recalled as algorithm 2.4 below. Next, we describe an algorithmic scheme (algorithm 2.5 below), slightly extending the Harker and Pang algorithm, with the goal of making it a framework accepting more ways of determining the stepsizes, in particular the one of Harker and Pang. It is for this last scheme that the lower bound on the iterative complexity is established.

The concepts of break-stepsizes and break-points will play a major part in the considered algorithms. After the definition of these notions, we clarify their connection with the nondifferentiability of F.

Definitions 2.1 (break-stepsize and break-point) A *break-stepsize* at $x \in \mathbb{R}^n$ along a direction $d \in \mathbb{R}^n$ is a real number $\check{\alpha} > 0$ such that there is an index $i \in [1\,{:}\,n]$ for which $x_i \neq (Mx+q)_i$ and $(x+\check{\alpha}d)_i = (Mx+q+\check{\alpha}Md)_i$. Then, $\check{x} := x+\check{\alpha}d$ is called a *break-point*.

Lemma 2.2 (kink of F at a break-point) *Let $\check{\alpha}$ be a break-stepsize at x along the direction d. Then F is not differentiable at $x + \check{\alpha}d$.*

Proof. Denote by $\check{x} := x + \check{\alpha}d$ the break-point corresponding to $\check{\alpha}$. Since $\check{\alpha}$ is a break-stepsize, there is an index $i \in [1\,{:}\,n]$ such that $x_i \neq (Mx + q)_i$ and $\check{x}_i = (M\check{x} + q)_i$, which implies that $d_i \neq (Md)_i$. Now an easy computation provides (see also [37])

$$F_i'(\check{x}; d) = \min(d_i, (Md)_i) \quad \text{and} \quad F_i'(\check{x}; -d) = \min(-d_i, -(Md)_i),$$

so that

$$F_i'(\check{x}; d) + F_i'(\check{x}; -d) = \min(d_i, (Md)_i) - \max(d_i, (Md)_i) < 0,$$

because $d_i \neq (Md)_i$. Hence F is nondifferentiable at $\check{x}$. □

Remark 2.3 Whilst F is nondifferentiable at a break-point, this is not necessary the case for Θ, as shown by the following example: $n = 1$, $M = 2$, $q = 0$, $x = -1$, and $d = 1$. Then $\check{\alpha} = 1$ is a break-stepsize since $-1 = x \neq Mx + q = -2$ and, for $\check{x} = x + \check{\alpha}d$, $\check{x} = M\check{x} + q = 0$. Since

$$F(x) = \begin{cases} 2x & \text{if } x \leqslant 0 \\ x & \text{otherwise} \end{cases} \qquad \text{and} \qquad \Theta(x) = \begin{cases} 2x^2 & \text{if } x \leqslant 0 \\ \frac{1}{2}x^2 & \text{otherwise,} \end{cases}$$

we see that F is nondifferentiable at $\check{x} = 0$, but that Θ is differentiable at the same point, which is agreement with the strong Fréchet differentiability of Θ at a zero of F, proved in [38; prop. 1]. □

This paper deals with the Newton-min algorithm [1], which is now described with more precision than in the introduction. The method has the flavor of a semismooth Newton method [39] for finding a zero of the nonsmooth function F defined by (1.2) [23]. At a point $x \in \mathbb{R}^n$, the indices in $[1:n]$ are partitioned in three subsets:

$$\begin{aligned} A_0(x) &:= \{i \in [1:n] : x_i < (Mx+q)_i\}, \\ E(x) &:= \{i \in [1:n] : x_i = (Mx+q)_i\}, \\ I_0(x) &:= \{i \in [1:n] : x_i > (Mx+q)_i\}. \end{aligned}$$

Since, for $i \in A_0(x) \cup I_0(x)$, F_i is differentiable at x, a Newton-like direction d should satisfy $F_i'(x)d = -F_i(x)$, which becomes $d_i = -x_i$ for $i \in A_0(x)$ and $M_{i:}d = -(Mx+q)_i$ for $i \in I_0(x)$, where $M_{i:}$ denotes the ith row of M. For $i \in E(x)$, F_i is usually nonsmooth at x; to reduce the size of the linear system to solve, these indices are dealt with like those in $A_0(x)$. In summary, the following index sets are introduced

$$A \equiv A(x) := A_0(x) \cup E(x), \qquad I \equiv I(x) := I_0(x), \tag{2.1}$$

and the *Newton-min direction* is defined by

$$d_A = -x_A \qquad \text{and} \qquad M_{I:}d = -(Mx+q)_I \equiv -M_{I:}x - q_I. \tag{2.2}$$

As a result, the point $\hat{x} := x + d$ targeted by the Newton-min algorithm satisfies

$$\hat{x}_A = 0 \qquad \text{and} \qquad (M\hat{x}+q)_I = 0. \tag{2.3}$$

The target point $\hat{x}$ is the one introduced by (1.3), since $\mathcal{S}(x) = I$ with the previous notation. The system (2.3) has a unique solution when M is nondegenerate, since its second identity also reads $M_{II}\hat{x}_I = -q_I$, which determines $\hat{x}_I = -M_{II}^{-1}q_I$ since then M_{II} is nonsingular.

The *plain Newton-min algorithm*, which takes $x^+ := \hat{x}$ as the iterate following the current one x, converges locally in one iteration when M is nondegenerate and x is in some neighborhood of a solution to the LCP [21]. It also converges globally if M is an **M**-matrix [1], but not if M is a **P**-matrix, since there are counter-examples in that case [4, 5, 6] (or even when M is a symmetric positive definite matrix [17]).

The purpose of the Harker and Pang algorithm [22; 1990, p. 275] is to improve the convergence properties of the plain Newton-min algorithm, as already mentioned in the introduction. For this, a stepsize $\alpha > 0$ is introduced along the Newton-min direction d, meaning that the iterate x^+ following x is computed by

$$x^+ = x + \alpha d.$$

The stepsize α has the very particular form

$$\alpha = \check{\alpha}_1 + \varepsilon_{\text{HP}},$$

where $\check{\alpha}_1$ is the first break-stepsize in $(0,1)$ at x along d and $\varepsilon_{\text{HP}} > 0$ is a positive number such that x^+ is not a break-point of F and $\Theta(x^+)$ is sufficienty smaller than $\Theta(x)$, in the sence that

$$\Theta(x^+) \leqslant (1 - 2\omega\alpha)\Theta(x), \tag{2.4}$$

for some $\omega \in (0, 1/2)$. Since, when $E(x) = \varnothing$ (this condition is satisfied recursively by all the iterates of the algorithm), the directional derivative of Θ at x along the Newton-min direction d takes the value $\Theta'(x; d) = -2\Theta(x)$ and the previous inequality is often referred to as the Armijo condition [2, 9]. This algorithm is summarized below. To the best of our knowledge, its global convergence has not been proved

Algorithm 2.4 (Newton-min-HP algorithm) It is supposed that the current iterate x is not a solution to (1.1) and verifies $E(x) = \varnothing$. The next iterate x^+ also verifies $E(x^+) = \varnothing$ and is computed as follows.

1. *Index sets.* Compute A and I by (2.1).
2. *Direction.* Compute the direction d by (2.2).
3. *Stepsize.* Compute the smallest break-stepsize $\check{\alpha}_1$, if any. Then, determine the stepsize $\alpha > 0$ by the following rules.
   - 3.1. If there is no break-stepsize in $(0,1)$, take $\alpha = 1$ and terminate with $x + d$,
   - 3.2. Otherwise take $\alpha = \check{\alpha}_1 + \varepsilon_{\text{HP}}$, where $\varepsilon_{\text{HP}} > 0$ is such that
     - 3.2.1. α is not a break-stepsize,
     - 3.2.2. (2.4) holds.
4. *New iterate.* $x^+ = x + \alpha d$.

It is not difficult to see that if the condition in step 3.1 holds, $\hat{x} := x + d$ is a solution to (1.1), which justifies the termination. This is because the inequalities verified by x are preserved at $\hat{x}$, since there is no break-point in the open segment $(x, \hat{x})$:

$$\hat{x}_A \leqslant (M\hat{x} + q)_A \qquad \text{and} \qquad \hat{x}_I \geqslant (M\hat{x} + q)_I. \tag{2.5}$$

Now, by (2.3), $\hat{x}_A = 0$ and $(M\hat{x} + q)_I = 0$, so that $0 \leqslant \hat{x} \perp (M\hat{x} + q) \geqslant 0$ (we have used (2.5) and $A \cup I = [1\,{:}\,n]$), meaning that $\hat{x}$ is a solution to the LCP.

The next algorithm is the one that is studied below. It differs from algorithm 2.4 by the way the stepsizes are determined along the Newton-min direction. Our goal in the design of algorithm 2.5 is not to make it efficient, but to make it as little binding as possible, in order to include as many variants of the Newton-min algorithm as possible. This way, the lower bound on its iterative complexity given in proposition 4.4 below will be valid for all the algorithms obeying the rules of algorithm 2.5.

Algorithm 2.5 (Newton-min-HP-ext scheme) It is supposed that the current iterate x is not a solution to (1.1) and verifies $E(x) = \varnothing$. The next iterate x^+ is then computed as follows.

1. *Index sets.* Compute A and I by (2.1).

2. *Direction.* Compute the direction d by (2.2).
3. *Stepsize.* Compute the two smallest distinct break-stepsizes $\check{\alpha}_1$ and $\check{\alpha}_2$, if any. Then, determine the stepsize $\alpha > 0$ by the following rules.
   3.1. If there is no break-stepsize in $(0, 1)$, take $\alpha = 1$ and terminate with $x + d$,
   3.2. If there is a single break-stepsize $\check{\alpha}_1$ in $(0, 1)$, take α in $(\check{\alpha}_1, 1]$,
   3.3. If there are at least two break-stepsizes $\check{\alpha}_1$ and $\check{\alpha}_2$ in $(0, 1)$, take α in $(\check{\alpha}_1, \check{\alpha}_2)$.
4. *New iterate.* $x^+ = x + \alpha d$.

Note that, in general, the Newton-min-HP algorithm is not a particular instance of algorithm 2.5, because it may occur that $\check{\alpha}_1 + \varepsilon_{\text{HP}} > \check{\alpha}_2$. Nevertheless, the scheme 2.5 includes the Newton-min-HP algorithm when $\varepsilon_{\text{HP}} > 0$ is sufficiently small and convergence of the iterates to a solution occurs. Indeed, when convergence occurs, it occurs in a finite number of iterations (by the above mentioned convergence in one step when the current iterate is in some neighborhood of a solution). Then the smallest value of $\check{\alpha}_2 - \check{\alpha}_1$ encoutered along the iterations (when both $\check{\alpha}_1$ and $\check{\alpha}_2$ exist) is > 0, implying that a sufficiently small positive ε_{HP} is in $(0, \check{\alpha}_2 - \check{\alpha}_1)$ or $\check{\alpha}_1 + \varepsilon_{\text{HP}} \in (\check{\alpha}_1, \check{\alpha}_2)$.

3 The Fathi problem

As claimed in the abstract, the lower bound on the iterative complexity of the Newton-min-HP-ext scheme is shown thanks to the Fathi problem. This LCP has its matrix formed with the one of the Murty LCP, which is first presented.

The Murty problem

The Murty problem [35] is often considered to have the following data M and q, and starting point x:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & \cdots & 0 \\ 2 & 2 & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \\ 2 & 2 & 2 & & 1 \end{pmatrix}, \qquad q = -e, \qquad \text{and} \qquad x = 0, \tag{3.1}$$

where e is the vector of all ones. Other values of q are considered in [36; chapter 6]. The matrix M is clearly a **P**-matrix (its principal minors have the value 1), so that the problem has a unique solution, which is $\bar{x} = (1, 0, \ldots, 0)$. This problem is extensively used for assessing algorithms [33, 22, 13], probably because some pivoting methods [34] are known to require an exponential number of iterations to solve it [36; theorem 6.4]. This problem is also relatively difficult to solve for the Newton-min algorithm, but not with the same severity [3].

The Fathi problem

In the Fathi problem [20; 1979], M, q, and the starting point are given by

$$M = \begin{pmatrix} 1 & 2 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & 6 & \cdots & 6 \\ 2 & 6 & 9 & 10 & \cdots & 10 \\ 2 & 6 & 10 & 13 & & 14 \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 2 & 6 & 10 & 14 & \cdots & 4(n-1)+1 \end{pmatrix}, \qquad q = -e, \qquad \text{and} \qquad x = 0. \tag{3.2}$$

Since $M = LL^{\mathsf{T}}$, where L is the nonsingular lower triangular Murty matrix [35], M is symmetric positive definite, hence a **P**-matrix. The unique solution to the Fathi problem is the same one as for the Murty problem, namely $\bar{x} = (1, 0, \ldots, 0)$. This problem was introduced in [20] to show the exponential iterative complexity of some pivot algorithms when the matrix of the LCP is symmetric positive definite.

The analysis of the Newton-min-HP-ext scheme below lies on the following two technical properties of the Fathi matrix. The first property determines the vector $v_I := M_{II}^{-1} e_I$, for some $I \subseteq [1\!:\!n]$, which, according to (2.3) and $q = -e$, are the nonzero components of the point $\hat{x}$ targeted by the Newton-min algorithm at any point x in $\{x \in \mathbb{R}^n : x_A < (Mx+q)_A$, and $x_I > (Mx+q)_I\}$, where A and I form a partition of $[1\!:\!n]$. In this lemma, the indices of the vector v_I are numbered with the indices in I. A similar convention is adopted for the matrices M_{II} and M_{AI}, where A is some other index subset of $[1\!:\!n]$.

Lemma 3.1 (two properties of the Fathi matrix) *Let $A := [2\!:\!k]$ for some $k \in [1\!:\!n]$ ($A = \varnothing$ if $k = 1$), $I := [1\!:\!n] \setminus A$, and e_A and e_I be the vectors of all ones, with indices taken in A and I respectively. Let M be the Fathi matrix given in (3.2). Then,*

1) *$v_I := M_{II}^{-1} e_I$ has its components, numbered by the indices of I, given by*

$$v_i = \begin{cases} \frac{2(2k-1)(n-k)+1}{4(k-1)(n-k)+1} & \text{if } i = 1, \\ (-1)^{i-k} \frac{2(n-i)+1}{4(k-1)(n-k)+1} & \text{if } i \in [k+1\!:\!n], \end{cases} \tag{3.3}$$

2) *$M_{AI} M_{II}^{-1} e_I > e_A$.*

Proof. 1) We provide a short verification proof. Let us denote by L the lower triangular matrix of Murty of dimension n, denoted M in (3.1), so that $M_{II} = L_{I:}(L_{I:})^{\mathsf{T}}$. We only have to check that the vector v_I given by formula (3.3) satisfies $L_{I:}(L_{I:})^{\mathsf{T}} v_I = e_I$.

Let us simplify the notation by introducing the positive numbers

$$a := \frac{2(n-k)}{4(k-1)(n-k)+1} \qquad \text{and} \qquad b := \frac{1}{4(k-1)(n-k)+1}$$

and let us show that $w := (L_{I:})^{\mathsf{T}} v_I$ takes the value (the numbers on the right are the

indices of the vector w)

$$w := \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ & 2 & 2 & \cdots & 2 \\ & \vdots & \vdots & \ddots & \vdots \\ & 2 & 2 & \cdots & 2 \\ & 1 & 2 & \ddots & \vdots \\ & & 1 & \ddots & 2 \\ & & & \ddots & 2 \\ & & & & 1 \end{pmatrix} v_I = \begin{pmatrix} 1 \\ -a \\ \vdots \\ -a \\ -b \\ b \\ \vdots \\ (-1)^{n-k}b \end{pmatrix}. \qquad \begin{matrix} 1 \\ 2 \\ \vdots \\ k \\ k+1 \\ k+2 \\ \vdots \\ n \end{matrix} \tag{3.4}$$

We will use the fact that, for $p \in \mathbb{N}$, there holds

$$1 - 3 + 5 - 7 + 9 + \cdots + (-1)^p(2p+1) = (-1)^p(p+1). \tag{3.5}$$

Let us first compute, for $j \in [k+1:n]$, the sum

$$\begin{aligned} \textstyle\sum_{i=j}^n v_i &= b \textstyle\sum_{i=j}^n (-1)^{i-k} \big[2(n-i)+1\big] \qquad [(3.3)] \\ &= b(-1)^{k-n} \textstyle\sum_{i=j}^n (-1)^{n-i} \big[2(n-i)+1\big] \\ &= b(-1)^{k-n} \big[1 - 3 + 5 - 7 + 9 + \cdots + (-1)^{n-j}[2(n-j)+1]\big] \\ &= b(-1)^{k-n} \big[(-1)^{n-j}(n+1-j)\big] \qquad [(3.5) \text{ with } p = n-j] \\ &= (-1)^{j-k} b(n+1-j). \end{aligned} \tag{3.6}$$

The rows of (3.4) with index in $[2:k]$ now follow from the previous computation with $j = k+1$, since

$$2 \textstyle\sum_{i=k+1}^n v_i = -2b(n-k) = -a. \tag{3.7}$$

The first row of (3.4) is also verified since by the previous computation

$$v_1 + 2 \textstyle\sum_{i=k+1}^n v_i = \big[2(2k-1)(n-k)+1\big]b - a = 1.$$

The last $(n-k)$ rows of (3.4) are also verified since for $j \in [k+1:n]$ there holds, using (3.3) and (3.6):

$$\begin{aligned} v_j + 2 \textstyle\sum_{i=j+1}^n v_i &= (-1)^{j-k} \,(2(n-j)+1)b + (-1)^{j+1-k} 2b(n-j) \\ &= (-1)^{j-k} b \,[2(n-j) + 1 - 2(n-j)] \\ &= (-1)^{j-k} b. \end{aligned}$$

It remains to observe that $L_{I:}w = e_I$, which follows from

$$L_{I:}w = \begin{pmatrix} 1 & & & & & & & \\ 2 & 2 & \cdots & 2 & 1 & & & \\ 2 & 2 & \cdots & 2 & 2 & 1 & & \\ \vdots & \vdots & \ddots & & \ddots & \ddots & \ddots & \\ 2 & 2 & \cdots & 2 & \cdots & 2 & 2 & 1 \end{pmatrix} w = \begin{pmatrix} 1 \\ 2 - 2(k-1)a - b \\ 2 - 2(k-1)a - 2b + b \\ \vdots \\ 2 - 2(k-1)a - 2b + 2b + \cdots + (-1)^{n-k}b \end{pmatrix} = e_I,$$

since $2(k-1)a+b=1$.

2) By the definitions of A and I, when $A \neq \varnothing$, M_{AI} has the form (the numbers on the right are the row indices of M_{AI})

$$M_{AI} = \begin{pmatrix} 2 & 6 & \cdots & 6 \\ 2 & 10 & \cdots & 10 \\ 2 & 14 & \cdots & 14 \\ \vdots & \vdots & & \vdots \\ 2 & 4k-2 & \cdots & 4k-2 \end{pmatrix} \begin{matrix} \\ 2 \\ 3 \\ 4 \\ \vdots \\ k \end{matrix}.$$

Since $M_{II}^{-1} e_I$ is the vector v_I given by (3.3), the row with index $i \in [2:k]$ of $M_{AI} M_{II}^{-1} e_I$ reads

$$2v_1 + (4i-2) \sum_{j=k+1}^{n} v_j = 2v_1 - (4i-2)(n-k)b,$$

where we have used (3.6) with $j = k+1$ (see also (3.7)). Its smallest value is obtained for the largest i, that is $i=k$, and is, thanks to (3.3):

$$\begin{aligned} 2v_1 - (4k-2)(n-k)b &= 2[2(2k-1)(n-k)+1]b - (4k-2)(n-k)b \\ &= [(4k-2)(n-k)+2]b \\ &> 1, \end{aligned}$$

which is the stated result. □

4 A lower bound on the iterative complexity

The goal of this section is to show that the Newton-min-HP-ext scheme (algorithm 2.5) converges in exactly n iterations on the instance of dimension n of the Fathi problem (3.2) when the algorithm starts at zero or in some neighborhood of zero. This gives a lower bound on the iterative complexity of the considered algorithmic scheme.

The proof of proposition 4.4 below consists in showing that, when the Newton-min-HP-ext scheme generates a sequence $\{x^k\}_{k\geqslant 0}$ with a starting point x^0 near zero (in the set $\mathcal{X}_1$ introduced below actually), there holds $x^k \in \mathcal{X}_{k+1}$, for $k \in [1:n]$, where $\mathcal{X}_k$ is defined by

$$\begin{aligned} \mathcal{X}_k := \Big\{ x \in \mathbb{R}^n :\ & x_{A_k} < (Mx+q)_{A_k} \text{ and } x_{I_k} > (Mx+q)_{I_k}, \\ & \tfrac{(Mx-e-x)_i}{(Mx-e-x)_{i+2}} < \tfrac{(M_{I_k I_k}^{-1} e_{I_k})_i}{(M_{I_k I_k}^{-1} e_{I_k})_{i+2}}, \text{ for all } i \in [k+1:n-2] \Big\}, \end{aligned} \tag{4.1}$$

with $A_k = [2:k]$ and $I_k := [2:k]^c$ (the complementary set of $[2:k]$ in $[1:n]$). In this definition, it is assumed that the integer interval $[i:j]$ is empty when $j<i$ (in particular, $A_1 = \varnothing$ and the strict inequalities after the second one are not present if $k \geqslant n-2$).

Remarks 4.1 1) There holds $0 \in \mathcal{X}_1$. Indeed, $A_1 = \varnothing$, $I_1 = [1:n]$, $0 > M0 + q = -e$ and, for $i \in [2:n-2]$:

$$\frac{(M0-e-0)_i}{(M0-e-0)_{i+2}} = 1 < \frac{2(n-i)+1}{2(n-i-2)+1} = \frac{(M^{-1}e)_i}{(M^{-1}e)_{i+2}},$$

where we have used (3.3). This observation also shows that $\mathcal{X}_1 \neq \varnothing$.

2) The fact that $\mathcal{X}_k \neq \varnothing$ for $k \in [2\,{:}\,n]$ will be a consequence of Lemma 4.2 below.

3) The set $\mathcal{X}_n = \{x \in \mathbb{R}^n : x_{[2\,:\,n]} < (Mx+q)_{[2\,:\,n]} \text{ and } x_1 > (Mx+q)_1\}$ is the one to which belongs the solution to the Fathi problem, namely $\bar{x} = (1, 0, \ldots, 0)$.

4) By the strict inequalities in their definition, the sets $\mathcal{X}_k$ are *open* (more precisely they are relative interiors of polyhedrons), so that they are not reduced to a single point. By the first two strict inequalities in their definition, these sets are also two by two *disjoint*.

5) The last group of inequalities in the definition (4.1) of $\mathcal{X}_k$ is not superfluous. For example, if $n = 4$, $x^0 := \frac{1}{31}e$ belongs to $\{x \in \mathbb{R}^4 : x > (Mx + q)\}$, which is the set $\mathcal{X}_1$ without the last inequality (corresponding to $i = 2$), but not to $\mathcal{X}_1$. Note also that with $x^0 = \frac{1}{31}e$, the next iterate x^1 satisfies $x^1_4 < (Mx^1 + q)_4$ and $x^1_{\{1,2,3\}} > (Mx^1 + q)_{\{1,2,3\}}$, so that x^1 is not in $\{x \in \mathbb{R}^4 : x_{A_2} < (Mx+q)_{A_2} \text{ and } x_{I_2} > (Mx+q)_{I_2}\}$, hence certainly not in $\mathcal{X}_2$. □

We start with the following fundamental lemma (fundamental for our purpose, since it contains the essential idea of the proof), which shows that if the current iterate x of the Newton-min-HP-ext scheme is in $\mathcal{X}_k$, the next iterate $x^+ = x + \alpha d$ will be in $\mathcal{X}_{k+1}$. In its proof, for positive integers i, s, and j, we use the notation

$$[i\,{:}\,s\,{:}\,j] := \{i, i + s, i + 2s, \ldots, i + \lfloor (j - i)/s \rfloor s\},$$

where $\lfloor\cdot\rfloor$ is the floor operator ($\lfloor r \rfloor$ is the integer number i such that r is in $[i, i+1)$); hence $[i\,{:}\,1\,{:}\,j] = [i\,{:}\,j]$.

Lemma 4.2 (one iteration from x to x^+) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Suppose that the current iterate x of the Newton-min-HP-ext scheme is in $\mathcal{X}_k$ for some $k \in [1\,{:}\,n-1]$. Then, the next iterate $x^+ = x + \alpha d$ is in $\mathcal{X}_{k+1}$ and, when $k \leqslant n-3$, the stepsize α is in $(0, 1)$.*

Proof. Let $k \in [1\,{:}\,n-1]$, $x \in \mathcal{X}_k$, and set $A \equiv A_k := [2\,{:}\,k]$ and $I \equiv I_k := [1\,{:}\,n] \setminus A$, so that

$$x_A < (Mx + q)_A \qquad \text{and} \qquad x_I > (Mx + q)_I.$$

The next iterate is then defined by $x^+ := x + \alpha d$, where

$$d = \begin{pmatrix} 0_A \\ M_{II}^{-1} e_I \end{pmatrix} - x \tag{4.2}$$

and the stepsize α is chosen as described in step 3 of algorithm 2.5. We have to prove that for some $\check{\alpha} \in (0, 1)$ there hold

$$(x + td)_A < (Mx + q + tMd)_A, \quad \text{for all } t \in [0, \check{\alpha}], \tag{4.3a}$$

$$(x + \check{\alpha} d)_{k+1} = (Mx + q + \check{\alpha} Md)_{k+1}, \tag{4.3b}$$

$$(x + td)_{I\setminus\{k+1\}} > (Mx + q + tMd)_{I\setminus\{k+1\}}, \quad \text{for all } t \in [0, \check{\alpha}], \tag{4.3c}$$

$$\text{if } k \leqslant n-3, \text{ then } \alpha \in (\check{\alpha}, 1), \tag{4.3d}$$

$$\frac{(Mx^+ - e - x^+)_i}{(Mx^+ - e - x^+)_{i+2}} < \frac{(M_{II}^{-1} e_I)_i}{(M_{II}^{-1} e_I)_{i+2}}, \quad \text{for all } i \in [k+2\,{:}\,n-2]. \tag{4.3e}$$

Indeed, if this is shown:

- by (4.3a)–(4.3c), the first break-stepsize $\check{\alpha}_1$ is $\check{\alpha} \in (0,1)$ and this break-stepsize is due to the index $k+1$,
- since $x \in \mathcal{X}_k$, it follows, using also (4.3b), that $(x+td)_{k+1} > (Mx+q+tMd)_{k+1}$ for $t < \check{\alpha}$, so that the reverse inequality holds for $t > \check{\alpha}$, implying that $k+1 \in I(x^+)$,
- since the stepsize α taken by algorithm 2.5 is less than the possible next break-stepsize $\check{\alpha}_2 > \check{\alpha}_1$, the inequalities (4.3a) and (4.3c) hold at $x+\alpha d = x^+$; hence $A(x^+) = [2, k+1]$ and $I(x^+) = [1\!:\!n] \setminus [2, k+1]$.
- Now with (4.3e), x^+ is in $\mathcal{X}_{k+1}$.

This implies that the first two strict inequalities in the definition of $\mathcal{X}_{k+1}$ hold. The last group of inequalities is just (4.3e). Finally, (4.3d) shows indeed that $\alpha \in (0,1)$. Let us now prove (4.3).

The equality $(x+td)_i = (Mx+q+tMd)_i$ is equivalent to $t(d-Md)_i = (Mx+q-x)_i$ or, using (4.2) and the value of $q = -e$, this identity can be rewritten

$$t\left[\begin{pmatrix} 0_A \\ M_{II}^{-1} e_I \end{pmatrix} - \begin{pmatrix} M_{AI} M_{II}^{-1} e_I \\ e_I \end{pmatrix} + Mx - x\right]_i = [Mx - e - x]_i \,. \tag{4.4}$$

Consider the indices $i \in A$. By (4.4), the equality $(x+td)_i = (Mx+q+tMd)_i$ is equivalent to

$$t\left[e_A - M_{AI} M_{II}^{-1} e_I + (Mx - e - x)_A\right]_i = [Mx - e - x]_i$$

or to

$$\frac{t}{1-t} = \frac{(Mx-e-x)_i}{(e_A - M_{AI} M_{II}^{-1} e_I)_i}.$$

Observe that the left-hand side is nonnegative if and ony if $t \in [0,1)$. Furthermore, the right-hand side is negative, since the numerator is positive by the assumption on x and the index $i \in A$, while the denominator is negative by point 2 of lemma 3.1. This implies that this identity cannot be realized by some $t \in [0,1]$. Consequently

$$\forall\, t \in [0,1]: \quad (x+td)_A < (Mx+q+tMd)_A$$

and (4.3a) holds, provided we show that $\check{\alpha} \leqslant 1$.

Consider now the indices $i \in I$. By (4.4), the equality $(x+td)_i = (Mx+q+tMd)_i$ is now equivalent to

$$t\left[M_{II}^{-1} e_I + (Mx - e - x)_I\right]_i = [Mx - e - x]_i$$

or to

$$\frac{t}{1-t} = \frac{(Mx-e-x)_i}{(M_{II}^{-1} e_I)_i}. \tag{4.5}$$

For $i \in I$, the numerator of the fraction in the right-hand side is negative, so that the right-hand side is nonnegative when $(M_{II}^{-1} e)_i$ is also negative, that is for $i \in [k+1\!:\!2\!:\!n]$ according to (3.3). By the monotonicity of the map $t \mapsto t/(1-t)$, the smallest break-stepsize at x along d is due to the index i giving the smallest fraction in the right-hand side. Since $x \in \mathcal{X}_k$, the third inequality in the definition (4.1) of $\mathcal{X}_k$ and the negativity of $(Mx-e-x)_i$ and $(M_{II}^{-1} e_I)_i$ for $i \in [k+1\!:\!2\!:\!n]$ tell us that this occurs for the smallest index $i \in [k+1\!:\!2\!:\!n]$, that is for $k+1$ (note that we use here only half of these third inequalities

in the definition of $\mathcal{X}_k$; the others will be used below for getting (4.3e)). Therefore, we have shown (4.3b) and (4.3c) for the unique solution $\check{\alpha}$ of

$$\frac{\check{\alpha}}{1-\check{\alpha}} = \frac{(Mx-e-x)_{k+1}}{(M_{II}^{-1}e_I)_{k+1}},$$

which is in $(0,1)$.

The reasonings in the previous two paragraphs also tell us that there are $\lceil (n-k)/2 \rceil$ break-stepsizes in the inteval $(0,1)$, which are due to the indices $[k+1:2:n]$. Therefore, when $k \leqslant n-3$, there are two break-stepsizes in $(0,1)$ and, by the step 3.3 of algorithm 2.5, there holds $\check{\alpha}_1 < \alpha < \check{\alpha}_2 < 1$, showing that α is in $(0,1)$. This shows (4.3d).

We still have to prove (4.3e) at the next iterate $x^+ = x + \alpha d$, where the stepsize $\alpha > 0$ is determined in step 3 of algorithm 2.5. Note that

$$\begin{aligned}(Mx^+ - e - x^+)_I &= (Mx-e-x)_I + \alpha(Md-d)_I \\ &= (Mx-e-x)_I + \alpha(e_I - (Mx)_I - M_{II}^{-1}e_I + x_I) \qquad [(4.2)] \\ &= (1-\alpha)(Mx-e-x)_I - \alpha M_{II}^{-1}e_I. \end{aligned} \tag{4.6}$$

Take now $i \in [k+2:n-2]$. Then $i \in I$, $k \leqslant n-4$, and $\alpha \in (0,1)$ by (4.3d). Using (4.6), the quotient in the LHS of (4.3e) becomes

$$\frac{(Mx^+ - e - x^+)_i}{(Mx^+ - e - x^+)_{i+2}} = \frac{-\alpha(M_{II}^{-1}e_I)_i + (1-\alpha)(Mx-e-x)_i}{-\alpha(M_{II}^{-1}e_I)_{i+2} + (1-\alpha)(Mx-e-x)_{i+2}}. \tag{4.7}$$

The quotient in the right-hand side can be written $\frac{a+s}{b+t}$ with the notation

$$\begin{aligned} a &:= -\alpha(M_{II}^{-1}e_I)_i, \qquad & s &:= (1-\alpha)(Mx-e-x)_i, \\ b &:= -\alpha(M_{II}^{-1}e_I)_{i+2}, \qquad & t &:= (1-\alpha)(Mx-e-x)_{i+2}. \end{aligned}$$

Since $t = (1-\alpha)(Mx-e-x)_{i+2} < 0$ (because $\alpha < 1$ and $i+2 \in I$), since $b+t = (Mx^+ - e - x^+)_{i+2} < 0$ (because $i+2 \in I(x^+) = [1:n] \setminus [2:k+1]$ by (4.3a)-(4.3c)), and since $\frac{s}{t} < \frac{a}{b}$ (by induction and because $\alpha \in (0,1)$), it follows that $\frac{a+s}{b+t} < \frac{a}{b}$. Therefore (4.7) becomes

$$\frac{(Mx^+ - e - x^+)_i}{(Mx^+ - e - x^+)_{i+2}} < \frac{-\alpha(M_{II}^{-1}e_I)_i}{-\alpha(M_{II}^{-1}e_I)_{i+2}} = \frac{(M_{II}^{-1}e_I)_i}{(M_{II}^{-1}e_I)_{i+2}},$$

which is (4.3e). □

By the previous lemma, if the initial iterate x^0 belongs to $\mathcal{X}_1$, after $n-1$ iterations, the iterate x^{n-1} belongs to

$$\mathcal{X}_n := \{x \in \mathbb{R}^n : x_{[2:n]} < (Mx+q)_{[2:n]} \text{ and } x_1 > (Mx+q)_1\}, \tag{4.8}$$

to which the solution $\bar{x} = e^1$ belongs. Hence the question arises to know whether one can have $x^{n-1} = \bar{x}$ and therefore converge in $n-1$ iterations. The next lemma invalidates this possibility.

Lemma 4.3 ($x^{n-1} \neq \bar{x}$) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.5, starting at a point $x \in \mathcal{X}_{n-1}$ finds a point $x^+ \in \mathcal{X}_n$ that differs from the solution $\bar{x} = e^1$ to the LCP problem (1.1).*

PROOF. Let us simplify the notation by setting $A := [2\,{:}\,n-1]$ and $I = \{1, n\}$. Then

$$\mathcal{X}_{n-1} = \{x \in \mathbb{R}^n : x_A < (Mx+q)_A \text{ and } x_I > (Mx+q)_I\}.$$

By algorithm 2.5, the iterate following $x \in \mathcal{X}_{n-1}$ satisfies

$$x^+ = x + \alpha((0_A, v_I) - x) = (1-\alpha)x + \alpha(0_A, v_I), \tag{4.9}$$

where v_I is given by (3.3) with the index set I introduced above (see the comment before lemma 3.1) and $\alpha > 0$ is the stepsize. We want to show that $x^+ \neq \bar{x}$.

We proceed by contradiction, assuming that $x^+ = \bar{x}$. Then, $x_1^+ = 1$, $x_A^+ = 0$, and $x_n^+ = 0$. According to (3.3), the first and third conditions read

$$(1-\alpha)x_1 + \alpha\,\frac{4n-5}{4n-7} = 1 \qquad \text{and} \qquad (1-\alpha)x_n + \alpha\,\frac{-1}{4n-7} = 0. \tag{4.10}$$

By the second identity in (4.10),

$$\alpha \neq 1. \tag{4.11}$$

Then (4.9) and $x_A^+ = 0$ imply that

$$x_A = 0. \tag{4.12}$$

Furthermore, adding the first identity in (4.10) and twice the second yields $(1-\alpha)(x_1 + 2x_n) + \alpha = 1$, which, thanks to (4.11), implies that

$$x_1 + 2x_n = 1. \tag{4.13}$$

Now, since $x \in \mathcal{X}_{n-1}$, there hold $x_1 > (Mx+q)_1$ and $x_n > (Mx+q)_n$. Therefore

$$\begin{aligned} x_1 + x_n &> (Mx+q)_1 + (Mx+q)_n \\ &= \big[x_1 + 2x_n - 1\big] + \big[2x_1 + (4n-3)x_n - 1\big] \qquad [(3.2) \text{ and } (4.12)] \\ &= 3x_1 + (4n-1)x_n - 2 \end{aligned}$$

or

$$(x_1 + 2x_n) + (2n-3)x_n < 1.$$

Using (4.13) and $n \geqslant 2$, we get $x_n < 0$, which is in contradiction with $\alpha \in [0,1]$ and the second identity in (4.10). □

The restriction on $n \geqslant 2$ in Lemma 4.3 is necessary, since when $n = 1$ the set $\mathcal{X}_{n-1}$ appearing in its statement does not exist.

Proposition 4.4 (worse case lower bound of the Newton-min-HP-ext scheme) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.5, starting at a point $x \in \mathcal{X}_k$, for some $k \in [1\,{:}\,n-1]$, finds the solution to the problem in $n-k+1$ iterations. In particular, when started at $x \in \mathcal{X}_1$ or at $x = 0$, algorithm 2.5 finds the solution in n iterations.*

PROOF. The first claim comes the fact that in one iteration the algorithm finds a point in $\mathcal{X}_{k+1}$ (by lemma 4.2). Applying this argument repetitively, we see that the algorithm finds a point on $\mathcal{X}_{n-1}$ in $n-k-1$ iterations. By lemma 4.3, the algorithm finds next a point in $\mathcal{X}_n$ in one more iteration, but this point is not the solution. Hence, one more iteration is necessary to get the solution and this is what algorithm 2.5 does. Indeed, if an iterate $x \in \mathcal{X}_n$, then there holds $x_{[2\,:\,n]} < (Mx+q)_{[2\,:\,n]}$ and $x_1 > (Mx+q)_1$ by (4.8), so that the next iterate x^+ satisfies $x^+_{[2\,:\,n]} = 0$ and $x^+_1 = 1$ if a unit stepsize is taken, which is indeed the choice of the algorithm. Hence x^+ is the solution.

The second claim comes from the fact that $0 \in \mathcal{X}_1$ (see remark 4.1(1)), hence a point in $\mathcal{X}_n$ is found in n iterations. □

Proposition 4.4 is not valid for $n = 1$. Indeed, in that case, $\mathcal{X}_1 = \mathcal{X}_n = \mathbb{R}$ and an initial iterate $x^0 \in \mathcal{X}_1$ can be the solution $\bar{x} = 1$, hence requiring no iteration to converge.

Corollary 4.5 (worse case lower bound of the Newton-min-HP algorithm) *Let M and q be the matrix and vector defining the Fathi problem (3.2) of dimension $n \geqslant 2$. Then, algorithm 2.4 with $\varepsilon_{\text{HP}} > 0$ sufficiently small, starting at a point $x \in \mathcal{X}_k$, for some $k \in [1\,{:}\,n-1]$, finds the solution to the problem in $n-k+1$ iterations. In particular, when started at $x \in \mathcal{X}_1$ or at $x = 0$, algorithm 2.4 finds the solution in n iterations.*

PROOF. This is because, when $\varepsilon_{\text{HP}} > 0$ is sufficiently small, the stepsizes α are in $(\check{\alpha}_1, \check{\alpha}_2)$ (see the comment given after the statement of algorithm 2.5) and proposition 4.4 applies. □

5 Numerical experiments

We have written a piece of software in Matlab, called Nmhp [19], which implements 3 methods.

(M_1) The first method is the Harker and Pang algorithm (algorithm 2.4), in which the extra stepsize $\varepsilon_{\text{HP}} > 0$ is determined from an initial value $\varepsilon^0_{\text{HP}} > 0$ prescribed by the user. In the numerical experiments reported below, we have taken the latter small ($\varepsilon^0_{\text{HP}} := 10^{-7}$ or 10^{-5}), while $\varepsilon_{\text{HP}} := \varepsilon^0_{\text{HP}}/2^i$, where i is the smallest nonnegative integer such that the two conditions in step 3.2 of algorithm 2.4 are satisfied. This is always possible since the number of break-stepsizes is finite and the Armijo condition (2.4) is satisfied with strict inequality for $\alpha = \check{\alpha}_1$ thanks to the choice of $\omega \in (0, 1/2)$.

(M_2) The second method is the extended version of the Harker and Pang algorithm (algorithm 2.5), in which the stepsize is fixed to $\alpha = (\check{\alpha}_1 + \check{\alpha}_2)/2$. According to lemma 4.2 on algorithm 2.5, the results would not be modified on the Fathi problem by taking any stepsize in $(\check{\alpha}_1, \check{\alpha}_2)$.

(M_3) The third method is a variant of the Newton-min algorithm with exact linesearch (meaning that $x^+ := x + \alpha d$ where $\alpha > 0$ is such that $\Theta(x^+) = \min\{\Theta(x + \alpha' d) : \alpha' > 0\}$). With exact linesearch, it is no longer guaranteed that $E(x) = \varnothing$ at all iterate x. This implies that a descent direction of Θ must be determined even when $E(x) \neq \varnothing$. We have chosen the Newton-min-hybrid direction defined in [3]. In this approach, an index i is chosen to be in $E(x)$ when $|x_i - (Mx+q)_i| \leqslant 10^{-11}$, it is in $A_0(x)$ when $x_i < (Mx+q)_i - 10^{-11}$, and in $I_0(x)$ when $x_i > (Mx+q)_i + 10^{-11}$.

These methods have been run on various instances of the Fathi problem, taking zero for initial iterate. The results (i.e., the number of iterations) are gathered in table 5.1, together

	Number of iterations				
	Harker-Pang algorithm			Algorithm 2.5 in Nmhp	Exact linesearch
n	In [22]	Algorithm 2.4 in Nmhp with $\varepsilon^0_{\text{HP}} = 10^{-7}$	$\varepsilon^0_{\text{HP}} = 10^{-5}$		
8	8	8	8	8	8
16	16	16	16	16	16
32	32	32	32	32	32
64	65	64	64	64	64
128	63	128	128	128	128
256	-	256	256	256	256
512	-	512	524	512	513
1024	-	1024	6367	1024	1025
2048	-	2048	16337	2048	2049

Table 5.1: Comparison of the number of iterations required to solve the Fathi problem of dimension n (1st column) starting at zero by several algorithms: the 2nd column gives the results of Harker and Pang in [22], the 3rd and 4th column gives the results of our implementation in Nmhp of algorithm 2.4 with $\varepsilon^0_{\text{HP}} = 10^{-7}$ and 10^{-5}, the 5th column are those of algorithm 2.5 in Nmhp, and the last column gives the results of the exact linesearch Newton-min-hybrid algorithm.

with those given by Harker and Pang in [22; table 5, example 2]. The first column gives the dimension n of the Fathi problem.

Here are some observations on the obtained results (see table 5.1).

(O_1) The results obtained by algorithm 2.5 of Nmhp (5th column) are in accordance with proposition 4.4: the number of iterations is n.

(O_2) The results given by Harker and Pang in [22] (2nd column) differ from n, for $n = 64$ and 128, and are not given for larger dimensions. The differences with algorithm 2.5 can only come from the stepsize $\alpha > 0$ taken along the Newton-min direction. The results of [22] for $n = 64$ and 128 could be explained by invoking rounding errors in the piece of software producing these results or, according to the proof of lemma 4.2, by the fact that $\check{\alpha}_1 + \varepsilon_{\text{HP}} > \check{\alpha}_2$ at some iterations when $n = 64$ and 128.

(O_3) Nevertheless, we have not been able to reproduce the results of Harker and Pang [22] with our implementation of algorithm 2.4: in accordance with corollary 4.5, when $\varepsilon^0_{\text{HP}}$ is sufficiently small one recovers the n iterations to find the solution ($\varepsilon^0_{\text{HP}} = 10^{-7}$ is small enough for the considered dimensions, see the 3rd column in table 5.1), but when $\varepsilon^0_{\text{HP}}$ is larger, the number of iterations has a tendency to increase (this is the case for $\varepsilon^0_{\text{HP}} = 10^{-5}$, see the 4th column in table 5.1), not to decrease as in the results of [22].

(O_4) The results obtained with the exact linesearch Newton-min-hybrid algorithm (last column) are surprising: the number of iterations differs from n by at most one unit. In other words, having a linesearch determining the best possible decrease of Θ does not improve the iteration counter (note that a modification of the stepsize changes the following direction). Proving this result would certainly be more difficult than the one shown in this paper, because the output of the code indicates that the change in the index sets (A, I) along the iterations does not follow the simple mechanism highlighted by lemma 4.2. Nevertheless, this last experiment supports the conclusion that any progress in the efficiency of the Newton-min is unlikely to come from a better linesearch procedure.

6 Conclusion

This paper is a contribution to the better understanding of the Newton-min algorithm with linesearch on the least-square merit function for solving the linear complementarity problem. It examines in details the behavior of the Harker and Pang globalization of the algorithm on the Fathi problem. It is mathematically proved and numerically observed that, if the first iterate is in some open polyhedral neighborhood of zero, then the algorithm requires exactly n iterations to find the solution to the problem (n is the number of variables). If this is not disastrous, for very large problems, it is not as attractive as the best path-following algorithms (interior or non-interior), whose iterative complexity is in $O(n^{1/2})$, and it does not reflect the excellent behavior of the Newton-min algorithm on many large scale problems coming from concrete applications [3]. Nevertheless, the realized precise computation of the number of iterations for the Fathi problem provides a lower bound on the provable iterative complexity of the Harker and Pang version of the Newton-min algorithm with linesearch, on a class of problems containing the Fathi problems. Numerical experiments suggest that this worse case lower bound could also be valid if an exact linesearch is performed.

References

[1] M. Aganagić (1984). Newton's method for linear complementarity problems. *Mathematical Programming*, 28, 349–362. [doi]. 18, 21

[2] L. Armijo (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16, 1–3. [journal]. 22

[3] I. Ben Gharbia, J.-P. Dussault, M. Frappier, J.Ch. Gilbert (2018). Globalization techniques of the Newton-min algorithm for solving the linear complementarity problem. Technical report. (to appear). 19, 23, 32, 33

[4] I. Ben Gharbia, J.Ch. Gilbert (2012). Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134, 349–364. [doi]. 18, 21

[5] I. Ben Gharbia, J.Ch. Gilbert (2013). An algorithmic characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3), 904–916. [doi]. 18, 21

[6] I. Ben Gharbia, J.Ch. Gilbert (2018). An algorithmic characterization of P-matricity II: adjustments, refinements, and validation. *SIAM Journal on Matrix Analysis and Applications* (submitted). [preprint]. 18, 21

[7] M. Bergounioux, M. Haddou, M. Hintermüller, K. Kunisch (2000). A comparison of a Moreau-Yosida-based active set strategy and interior point methods for constrained optimal control problems. *SIAM Journal on Optimization*, 11, 495–521. [doi]. 18

[8] M. Bergounioux, K. Ito, K. Kunisch (1999). Primal-dual strategy for constrained optimal control problems. *SIAM Journal on Control and Optimization*, 37, 1176–1194. [doi]. 18

[9] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (2006). *Numerical Optimization – Theoretical and Practical Aspects* (second edition). Universitext. Springer Verlag, Berlin. [authors] [editor] [doi]. 18, 22

[10] J.V. Burke, S. Xu (1999). A polynomial time interior-point path-following algorithm for LCP based on Chen-Harker-Kanzow smoothing techniques. *Mathematical Programming*, 86(1), 91–103. [doi]. 19

[11] J.V. Burke, S. Xu (2002). Complexity of a noninterior path-following method for the linear complementarity problem. *Journal of Optimization Theory and Applications*, 112(1), 53–76. [doi]. 19

[12] R. Chandrasekaran (1970). A special case of the complementary pivot problem. *Opsearch*, 7, 263–268. 18

[13] B. Chen, P.T. Harker (1993). A non-interior continuation method for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 14, 1168–1190. [doi]. 19, 23

[14] S.J. Chung (1989). NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60, 393–399. [doi]. 18

[15] A.R. Conn, N.I.M. Gould, Ph.L. Toint (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization 1. SIAM and MPS, Philadelphia. [doi]. 18

[16] R.W. Cottle, J.-S. Pang, R.E. Stone (2009). *The Linear Complementarity Problem*. Classics in Applied Mathematics 60. SIAM, Philadelphia, PA, USA. 18

[17] F.E. Curtis, Z. Han, D.P. Robinson (2015). A globally convergent primal-dual active-set framework for large-scale convex quadratic optimization. *Computational Optimization and Applications*, 60(2), 311–341. [doi]. 18, 21

[18] J.E. Dennis, R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs. 18

[19] J.-P. Dussault, M. Frappier, J.Ch. Gilbert (2018, 6). Nmhp, version 1.0. https://who.rocq.inria.fr/Jean-Charles.Gilbert/codes/nmhp/nmhp.zip. 31

[20] Y. Fathi (1979). Computational complexity of LCPs associated with positive definite symmetric matrices. *Mathematical Programming*, 17, 335–344. 19, 24

[21] A. Fischer, C. Kanzow (1996). On finite termination of an iterative method for linear complementarity problems. *Mathematical Programming*, 74, 279–292. [doi]. 18, 21

[22] P.T. Harker, J.-S. Pang (1990). A damped-Newton method for the linear complementarity problem. In E.L. Allgower, K. Georg (editors), *Computational Solution of Nonlinear Systems of Equations*, Lecture in Applied Mathematics 26. AMS, Providence, RI. 18, 19, 20, 21, 23, 32, 33

[23] M. Hintermüller, K. Ito, K. Kunisch (2003). The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13, 865–888. [doi]. 18, 21

[24] K. Hotta, M. Inaba, A. Yoshise (2000). A complexity analysis of a smoothing method using CHKS-functions for monotone linear complementarity problems. *Computational Optimization and Applications*, 17(2-3), 183–201. [doi]. 19

[25] A.F. Izmailov, M.V. Solodov (2014). *Newton-Type Methods for Optimization and Variational Problems*. Springer Series in Operations Research and Financial Engineering. Springer. 18

[26] Ch. Kanzow (1996). Some noninterior methods for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 17(4), 851–868. [doi]. 19

[27] Ch. Kanzow (2004). Inexact semismooth Newton methods for large-scale complementarity problems. *Optimization Methods and Software*, 19, 309–325. [doi]. 18

[28] M. Kojima, N. Megiddo, T. Noma, A. Yoshise (1991). *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*. Lecture Notes in Computer Science 538. Springer, Berlin. [doi]. 18, 19

[29] M. Kojima, N. Megiddo, T. Noma, A. Yoshise (1991). A unified approach to interior point algorithms for linear complementarity problems: A summary. *Operations Research Letters*, 10(5), 247–254. [doi]. 19

[30] M. Kojima, S. Shindo (1986). Extension of Newton and quasi-Newton methods to systems of PC^1 equations. *Journal of Operations Research Society of Japan*, 29, 352–375. 18

[31] M.M. Kostreva (1976). *Direct algorithms for complementarity problems*. PhD Thesis, Rensselaer Polytechnic Institute, Troy, New York. 18

[32] O.L. Mangasarian (1977). Solution of symmetric linear complementarity problems by iterative methods. *Journal of Optimization Theory and Applications*, 22, 465–485. [doi]. 18

[33] S. Mizuno, A. Yoshise, T. Kikuchi (1989). Practical polynomial time algorithms for linear complmentarity problems. *Journal of Operations Research Society of Japan*, 32(1), 75–92. [doi]. 23

[34] K.G. Murty (1974). Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, 11, 123–130. 23

[35] K.G. Murty (1978). Computational complexity of complementarity pivot methods. *Mathematical Programming Study*, 7, 61–73. 23, 24

[36] K.G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming* (Internet edition, prepared by Feng-Tien Yu, 1997). Heldermann Verlag, Berlin. 23

[37] J.-S. Pang (1990). Newton's method for B-differentiable equations. *Mathematics of Operations Research*, 15, 311–341. [doi]. 19, 20

[38] J.-S. Pang (1991). A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems. *Mathematical Programming*, 51, 101–131. [doi]. 21

[39] L. Qi, J. Sun (1993). A nonsmooth version of Newton's method. *Mathematical Programming*, 58, 353–367. [doi]. 18, 21

[40] H. Samelson, R.M. Thrall, O. Wesler (1958). A partition theorem for the Euclidean n-space. *Proceedings of the American Mathematical Society*, 9, 805–807. [editor]. 18

CHAPITRE 3

Globalisation convergente de la méthode de Newton semi-lisse

Résumé

Cet article introduit l'algorithme Newton-min-hybrid qui est une variante apportée à la méthode de Newton semi-lisse appliqué au problème de complémentarité. Le but est de présenter un algorithme qui n'a pas besoin de résoudre des sous-problèmes de complémentarité linéaire puisqu'à ce jour tous les algorithmes connus de cette famille peuvent être contraints à le faire. Dans le cas de l'algorithme de B-Newton par exemple cela ne fait pas de sens lorsque le problème initial est déjà un problème linéaire. L'algorithme Newton-min-hybrid n'a besoin que de résoudre des systèmes d'équations linéaires et parfois des problèmes d'optimisation quadratique convexe ce qui est plus simple en général. D'abord, on présente en toute généralité la méthode de Newton semi-lisse et les méthodes de B-Newton et de Han, Pang et Rangaraj qui ont inspiré Newton-min-hybrid. Ensuite, on présente Newton-min-convergent une variante de Newton-min qui assure la convergence mais qui n'est pas si efficace en pratique à cause de son besoin de résoudre souvent des

problèmes d'optimisation de grande taille. Pour éviter les itérations coûteuses, une autre méthode, Newton-min-descent est présentée. Cet algorithme ne résout que très rarement des problèmes d'optimisation en pratique. Ces sous-problèmes résolus sont généralement de très petite taille. Par contre, il ne semble toutefois pas convergent. Des exemples où la suite des itérés semble converger vers un point d'accumulation qui n'est pas une solution ont été trouvés. Par contre, il trouve toujours une direction de descente. Ainsi, on propose un algorithme hybride qui utilise les deux stratégies précédentes. La direction de descente moins coûteuse est priorisée sauf si on détecte qu'il y a un risque de converger vers un point qui n'est pas la solution. On termine avec la présentation de résultats qui comparent Newton-min-hybrid avec d'autres algorithmes. Ces résultats démontrent que la méthode introduite peut être très efficace pour résoudre des problèmes tirés de vraies applications et mérite encore d'être étudiée.

Commentaires

Dans le cadre de ce travail, j'ai agi comme auteur principal de l'article. J'ai d'abord implémenté en Matlab quelques variantes de l'algorithme Newton-min dont Newton-min-descent qui assure une direction de descente. J'ai aussi exploré quelques alternatives comme la recherche linéaire exacte, l'interpolation quadratique ou la recherche linéaire non-monotone. Les résultats des deux premières options n'étaient pas convaincants tandis que la recherche non-monotone a été laissée de côté pour l'instant à cause de la difficulté supplémentaire que cela apporterait à une preuve de convergence. Une fois la preuve de convergence globale de Newton-min-hybrid trouvée, j'ai implémenté l'algorithme tel que décrit à la section 4 de l'article. J'ai aussi effectué plusieurs tests qui ne sont pas mentionnés dans l'article afin de trouver quels paramètres semblent donner les meilleurs résultats. Finalement, j'ai réussi à alléger les hypothèses nécessaires au théorème de convergence globale de Newton-min-hybrid pour arriver à celles données dans l'article.

Cet article en cours de finition sera soumis à SIAM Journal on Optimization.

Globally Convergent Modification of the Semismooth Newton Method Applied to Complementarity Problems

J.-P. Dussault [†], M. Frappier [‡], and J. Ch. Gilbert [§]

The semismooth Newton method is a very efficient approach for computing a zero of a large class of nonsmooth functions. When the current iterate is sufficiently close to a solution and the function is strongly semismooth, the method converges to the zero quadratically. Otherwise, it is difficult to force the convergence using line search or trust regions because a semismooth Newton direction is not necessarily a descent direction of the associated least-square function, unlike when the function is differentiable. We explore this question in the particular case of a nonsmooth equational formulation of the complementarity problem using the min function. We propose a globally convergent algorithm using a modification of the semismooth Newton direction that makes it a descent direction of the least-square function at any point. Next, we present heuristics to improve its efficiency while maintaining the local finite termination of the algorithm in a neighborhood of a solution when the function is linear. An intensive numerical exploration of a careful implementation of the proposed method shows that it is competitive on some applications and on randomly generated problems, which we view as an indication saying that the approach deserves more investigations.

Keywords: finite convergence, global convergence, least-square merit function, complementarity problem, line search, Newton-min algorithm, P-matrix, semismooth Newton.

1 Introduction

Let $n \in \mathbb{N}^+$, $x \in \mathbb{R}^n$ and a given function $f : X \to \mathbb{R}^n$ defined on a subset $X \in \mathbb{R}^n$. The complementarity problem is to find a vector x such that

$$x \geqslant 0, \quad f(x) \geqslant 0, \quad x^{\mathsf{T}} f(x) = 0, \tag{1.1}$$

where the inequalities must be understood component wise. The term complementarity comes from the fact that, for all $i \in [1\,{:}\,n]$, either x_i or $f_i(x)$ must vanish; conditions that may be realized in 2^n different ways. The complementarity problems arise naturally in the optimality conditions of Karush, Kuhn and Tucker in optimization problems. Indeed, when $f(x) = \nabla g(x)$ the solution of this complementarity problem is the KKT conditions of $\min g(x)$ subject to $x \geqslant 0$. Complementarity problems are often used to model problems described by several systems of equations which are in some way in competition. The one which is active in a given place and time, corresponding to a common index of x and $f(x)$ depends on thresholds which are reached or not. If the threshold $x_i = 0$ is not reached,

[†]Département d'Informatique, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: Jean-Pierre.Dussault@Usherbrooke.ca.

[‡]Département de Mathématiques, Faculté des Sciences, Université de Sherbrooke, Canada. E-mail: Mathieu.Frappier@usherbrooke.ca.

[§]INRIA Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France. E-mail: Jean-Charles.Gilbert@inria.fr.

i.e., $x_i > 0$, then the equation $f_i(x) = 0$ is active. Examples of problems modeled by complementarity are numerous. The problems of contact, the problems of appearance and disappearance of phases in the multiphase flows, the problems of precipitation-dissolution in chemistry, meteorology and economic equilibrium are a few examples.

Several ways to solve (1.1) have been explored since it was introduced by Cottle in his Ph.D thesis in 1964. One of them is to consider the minimum function reformulation $F : \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$F(x) = \min(x, f(x)). \tag{1.2}$$

Indeed, when a point x^* is such as $F(x^*) = 0$, it is not hard to see that it solves (1.1). This reformulation is not differentiable but the directional derivative of F exists for every directions throughout the space and has the semismoothness property in the sense of Mifflin [28] when f is continuously differentiable. This transformed problem may now be solved by a semismooth Newton method. This method [35] is a variant of the Newton method for solving a system of equation $F(x) = 0$ in x, where $F : \mathbb{E} \to \mathbb{F}$ is not Fréchet differentiable ($\mathbb{E}$ and $\mathbb{F}$ are vector spaces, usually of the same finite dimension), but has the so-called semismoothness property. The next iterate, say x^+, is computed from the current one, say x, by the recurrence $x^+ = x + d$, where d solves $F(x) + Jd = 0$ for some generalized Jacobian J. This operator J is a element of the Clarke differential $\partial_C F(x)$ [11] or a surrogate of such an element when its computation is too difficult or impossible (see [20, 26] for the projection on a convex polyhedron).

An algorithm using this strategy to solve a complementarity problem is the Newton method for B-differentiable equations presented by Pang in 1990 [33] and discussed again in [34]. This algorithm computes a direction d^{B} which is the solution of $F(x) + BF(x)d^{\mathrm{B}} = 0$, where $BF(x)$ is the B-derivative of F at x. It is shown that d^{B} is a descent direction of the associated merit function which is the function $\Theta : \mathbb{R}^n \to \mathbb{R}$ defined at $x \in \mathbb{R}^n$ by

$$\Theta(x) := \frac{1}{2}\|F(x)\|^2 = \frac{1}{2}\|\min(x, f(x))\|^2. \tag{1.3}$$

Then the next iterate is then $x + \alpha d$ with α a step size ensuring the sufficient decrease of Θ. Actually, by defining the index sets

$$\begin{array}{rcl} A \equiv A(x) & := & \{i : x_i < f_i(x)\}, \\ I \equiv I(x) & := & \{i : x_i > f_i(x)\}, \\ E \equiv E(x) & := & \{i : x_i = f_i(x)\}, \end{array} \tag{1.4}$$

the B-Newton direction is obtained by solving the following mixed linear complementarity problem in the variable d^{B}:

$$\left\{ \begin{array}{l} (x + d^{\mathrm{B}})_A = 0 \\ f_I(x) + \nabla f_I(x) d^{\mathrm{B}} = 0 \\ 0 \leqslant (x + d^{\mathrm{B}})_E \perp (f_E(x) + \nabla f_E(x) d^{\mathrm{B}} \geqslant 0. \end{array} \right. \tag{1.5}$$

When f is nonlinear, it is reasonable to think that the linear subproblem is easier than the main problem itself. However, a counterpart is the particular case when it is applied on linear complementarity problem (LCP). An example of a situation where problem (1.5) is exactly the original problem is shown in [4]. Moreover, the convergence theorem of this

algorithm uses the assumption that $\limsup \alpha^k > 0$ which is a weak result since this strong assumption relates to the algorithm rather than the problem.

A refinement of this method is the Han, Pang and Rangaraj algorithm presented in [21]. The functioning of this algorithm is given a point x, a direction d is computed by solving the equation $F(x) + G(x; d) = 0$, where $G : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$G_i(x, d) = \begin{cases} d_i & \text{if } x_i < f_i(x),\ f_i(x) \geqslant 0 \\ \nabla f_i(x)^\mathsf{T} d & \text{if } x_i > f_i(x),\ x_i \geqslant 0 \\ \min(d_i, \nabla f_i(x)^\mathsf{T} d) & \text{otherwise.} \end{cases} \tag{1.6}$$

Then a line search is performed to obtain the next iterate. Note that like the B-Newton method, it may be necessary to solve an LCP for indices in the last case of G for each iteration. This brings the same problem as before and the global convergence when f is linear cannot be ensured. Some other globalization techniques explored projections or a path search (see [37, 36, 15]) but to the best of our knowledge, all the globalization techniques of the semismooth Newton method may have to solve an LCP subproblem.

The goal of this paper is to present a variation of the Newton method with the global convergence property as well but instead of solving LCP subproblems, it minimizes a constrained convex quadratic function which can be done in polynomial time. Moreover, the introduced method Newton-min-convergent can face the presence of kinks on Θ. This least-square function is natural, since it has been used, often with success, for globalizing the Newton method when the function F is smooth [14, 12, 8, 23]. Newton-min-convergent always compute a descent direction of Θ which makes it possible to force the progress towards the solution by line search. In section 4, an hybrid algorithm derived from Newton-min-convergent using heuristics to reduce the number of quadratic subproblems to solve and their dimension is presented. This algorithm, Newton-min-hybrid is proved to be well defined, globally convergent as well and when f is linear, it converges in one step in a neighborhood of the solution. Next, it is compared numerically with Pathlcp and Minmap-Newton on a large set of problems, including those of Murty, Fathi, Harker and Pang. These tests show that the method is very competitive and often it does not need to solve quadratic subproblems at all.

Notation. For the $n \times n$ matrix M and index sets I and $J \subseteq [1\,{:}\,n]$, we denote by M_{IJ} the submatrix of M formed of its elements with row indices in I and column indices in J. We also define $M_{I:} := M_{I[1\,:\,n]}$ and $M_{II}^{-1} := (M_{II})^{-1}$.

2 Newton-min-convergent

The algorithm presented in this section is inspired by the one of Han, Pang and Rangaraj [21] and B-Newton. It has a descent direction even on the kinks of the least square merit function Θ and the progress towards the solution is provided by line search. To simplify what follows, the index sets (1.4) will be regarded separately according to the sign

of x and $f(x)$. Now consider

$$\begin{array}{rcl} A^+ \equiv A^+(x) & := & \{i : x_i < f_i(x),\ f_i(x) \geqslant 0\}, \\ A^- \equiv A^-(x) & := & \{i : x_i < f_i(x),\ f_i(x) < 0\}, \\ I^+ \equiv I^+(x) & := & \{i : x_i > f_i(x),\ x_i \geqslant 0\}, \\ I^- \equiv I^-(x) & := & \{i : x_i > f_i(x),\ x_i < 0\}, \\ E^+ \equiv E^+(x) & := & \{i : x_i = f_i(x) \geqslant 0\}, \\ E^- \equiv E^-(x) & := & \{i : x_i = f_i(x) < 0\}. \end{array} \tag{2.1}$$

Moreover, E_A and E_I denote an arbitrary partition of E^+.

2.1 Newton-min-convergent direction

Newton-min-convergent uses the same function G defined at (1.6) but unlike the Newton method, where $F(x) + G(x; d)$ has to be 0 in the computation of the direction, it allows values greater than for the indices $i \in (A^- \cup I^- \cup E^-)$. Similarly the direction computed by this method has to be a vector satisfying the following equations

$$\left\{ \begin{array}{l} (x+d)_{A^+ \cup E_A} = 0 \\ (f(x) + \nabla f(x)d)_{I^+ \cup E_I} = 0 \\ (x+d)_{(A^- \cup I^- \cup E^-)} \geqslant 0 \\ (f(x) + \nabla f(x)d)_{(A^- \cup I^- \cup E^-)} \geqslant 0. \end{array} \right. \tag{2.2}$$

Our claim is that any d satisfying (2.2) is a descent direction of Θ. First, recall the map Θ is directionally differentiable as a composition of directionally differentiable and Lipschitz continuous functions ([8; lemma 11.1] for example). Furthermore, the chain rule applies:

$$\Theta'(x; d) = F(x)^{\mathsf{T}} F'(x; d). \tag{2.3}$$

Proposition 2.1 *Let $d \in \mathbb{R}^n$ such as d satisfies (2.2) then*

$$\Theta'(x; d) \leqslant -2\Theta(x). \tag{2.4}$$

Proof. A straightforward computation yields

$$F_i'(x; d) = \left\{ \begin{array}{ll} d_i & \text{if } i \in A \\ \nabla f_i(x)d & \text{if } i \in I \\ \min(d_i, \nabla f_i(x)d) & \text{if } i \in E \end{array} \right.$$

so that by the formula (2.3) of $\Theta'(x; d)$ and $x_E = f_E(x)$ there holds

$$\begin{array}{rcl} \Theta'(x; d) & = & x_{A^+}^{\mathsf{T}} d_{A^+} + f_{I^+}(x)^{\mathsf{T}} \nabla f_{I^+}(x)d + x_{A^-}^{\mathsf{T}} d_{A^-} + f_{I^-}(x)^{\mathsf{T}} \nabla f_{I^-}(x)d \\ & & + x_E^{\mathsf{T}} \min(d_E, \nabla f_E(x)d) \\ & = & -\|x_{A^+}\|^2 - \|f_{I^+}(x)\|^2 + x_{A^-}^{\mathsf{T}} d_{A^-} + f_{I^-}(x)^{\mathsf{T}} \nabla f_{I^-}(x)d \\ & & + x_E^{\mathsf{T}} \min((x+d)_E, (f(x) + \nabla f(x)d)_E) - \|x_E\|^2 \\ & \geqslant & -\|x_{A^+}\|^2 - \|f_{I^+}(x)\|^2 - \|x_{A^-}\|^2 - \|f_{I^-}(x)\|^2 - \|x_E\|^2 \\ & & + x_E^{\mathsf{T}} \min((x+d)_E, (f(x) + \nabla f(x)d)_E) \\ & = & -2\Theta(x) + x_E^{\mathsf{T}} \min((x+d)_E, (f(x) + \nabla f(x)d)_E) \end{array}$$

Now, It suffices to show that the second term in the right-hand side is nonpositive. Let $i \in E$.

- When $i \in E^+$, (2.2) shows that one of the two values $(x+d)_i$ and $(f(x)+\nabla f(x)d)_i$ is zero so that their minimum is nonpositive, resulting in $x_i \min((x+d)_i, (f(x)+\nabla f(x)d)_i) \leqslant 0$.
- When $i \in E^-$, (2.2) shows that the two values $(x+d)_i$ and $(f(x)+\nabla f(x)d)_i$ are nonnegative, so that their minimum is also nonnegative, resulting in $x_i \min((x+d)_i, (f(x)+\nabla f(x)d)_i) \leqslant 0$.

□

Before stating a sufficient condition so that equations (2.2) always admit at least one solution, consider the the following definition.

Definition 2.2 *A square matrix M for which a vector x satisfying*

$$Mx > 0 \qquad \textit{and} \qquad x \geqslant 0 \tag{2.5}$$

*exists is called an **S**-matrix.*

Proposition 2.3 *If f satisfies the two following assumptions then the equations (2.2) admit at least one solution.*

Assumption 1 *The function f is differentiable and its jacobian ∇f is such as ∇f_{II} is nonsingular for any subset I of $[1:n]$.*

Assumption 2 *The Schur complement $\nabla f_{\tilde{E}\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{I}}(x)(\nabla f_{\tilde{I}\tilde{I}}(x))^{-1}\nabla f_{\tilde{I}\tilde{E}}(x)$ where $\tilde{E} = A^- \cup I^- \cup E^-$ and $\tilde{I} = I^+ \cup E_I$ is an **S**-matrix for any subset $\tilde{I}$ and $\tilde{E}$ of $[1:n]$.*

Proof. By writing

$$d_{\tilde{I}} = -\nabla f_{\tilde{I}\tilde{I}}^{-1}(x)(f_{\tilde{I}}(x) + \nabla f_{\tilde{I}\tilde{A}}(x)d_{\tilde{A}} + \nabla f_{\tilde{I}\tilde{E}}(x)d_{\tilde{E}}) \tag{2.6}$$

where $\tilde{A} = A^+ \cup E_A$ and by replacing $d_{\tilde{I}}$ in the last equation of (2.2) with (2.6), we obtain:

$$\begin{aligned}
0 \leqslant\ & f_{\tilde{E}}(x) + \nabla f_{\tilde{E}\tilde{A}}(x)d_{\tilde{A}} + \nabla f_{\tilde{E}\tilde{I}}(x)d_{\tilde{I}} + \nabla f_{\tilde{E}\tilde{E}}(x)d_{\tilde{E}} \\
=\ & f_{\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{A}}(x)x_{\tilde{A}} + \nabla f_{\tilde{E}\tilde{I}}(x)(-\nabla f_{\tilde{I}\tilde{I}}^{-1}(x)(f_{\tilde{I}}(x) - \nabla f_{\tilde{I}\tilde{A}}(x)x_A + \nabla f_{\tilde{I}\tilde{E}}(x)d_{\tilde{E}})) \\
& + \nabla f_{\tilde{E}\tilde{E}}(x)d_{\tilde{E}} \\
=\ & (\nabla f_{\tilde{E}\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}(x)\nabla f_{\tilde{I}\tilde{E}}(x))d_{\tilde{E}} + f_{\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{A}}(x)x_{\tilde{A}} \\
& - \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}f_{\tilde{I}}(x) + \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}\nabla f_{\tilde{I}\tilde{A}}(x)x_{\tilde{A}}.
\end{aligned}$$

To lighten the text, consider the following changes of variables

$$M = \nabla f_{\tilde{E}\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}(x)\nabla f_{\tilde{I}\tilde{E}}(x)$$

$$q = f_{\tilde{E}}(x) - \nabla f_{\tilde{E}\tilde{A}}(x)x_{\tilde{A}} - \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}f_{\tilde{I}}(x) + \nabla f_{\tilde{E}\tilde{I}}(x)\nabla f_{\tilde{I}\tilde{I}}^{-1}\nabla f_{\tilde{I}\tilde{A}}(x)x_{\tilde{A}}.$$

Since $Md_{\tilde{E}} + q \geqslant 0$, it follows that $M(x+d_{\tilde{E}}) - Mx + q \geqslant 0$ and since M is an **S**-matrix, by [13; proposition 3.1.5] the LCP $0 \leqslant (x+d_{\tilde{E}}) \perp (M(x+d_{\tilde{E}}) - Mx + q) \geqslant 0$ admits at least one solution. □

Note that assumptions 1 and 2 are always verified when ∇f is a $\mathbf{P}$-matrix (i.e. all principal minor are strictly positive). Furthermore, d satisfying (2.2) is not unique as soon as $(A^- \cup I^- \cup E^-) \neq \varnothing$ which can occurs frequently. That is why a choice has to be made and the most natural one is to take the direction of minimal norm. Hence the Newton-min-convergent direction is the solution of the quadratic optimization problem

$$\begin{cases} \min_d \ \frac{1}{2}\|d\|^2 \\ d \text{ satisfies (2.2).} \end{cases} \tag{2.7}$$

Thus it may be necessary to solve a convex quadratic optimization problem at each iteration, which can be done with a polynomial in n complexity. Since the set defined by (2.2) is a convex polyhedron (hence closed), problem (2.7) has a unique solution as soon as the system (2.2) is feasible which is always the case thanks to assumptions 1 and 2. Note that the solution to (2.7) may not be complementary in the sense that it may happen that $(x+d)^\mathsf{T} f(x+d)$ is nonzero. Indeed, a complementary solution of (2.2) is the B-Newton direction but may be hard to compute.

Counter-example 2.4 (d is not complementary) Consider the particular instance of problem $0 \leqslant x \perp (Mx+q) \geqslant 0$, in which

$$n = 2, \qquad M = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \qquad q = \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \qquad \text{and} \qquad x = \begin{pmatrix} -1 \\ -1 \end{pmatrix}. \tag{2.8}$$

Note that M is a $\mathbf{P}$-matrix so this problem has only one solution [32]. Since $Mx+q = (-4,-1)^\mathsf{T}$, it follows that $I^- = \{1\}$ and $E^- = \{2\}$. Hence problem (2.7) reads

$$\begin{cases} \min_{d\in\mathbb{R}^2} \ \frac{1}{2}\|d\|^2 \\ d_1 + d_2 \geqslant 4 \\ d_1 + 2d_2 \geqslant 1 \\ d_1, d_2 \geqslant 1 \end{cases} \qquad \text{or} \qquad \begin{cases} \min_{d\in\mathbb{R}^2} \ \frac{1}{2}\|d\|^2 \\ d_1 + d_2 \geqslant 4 \\ d_1, d_2 \geqslant 1 \end{cases} \tag{2.9}$$

since the second inequality is redundant. Its solution $d = (2,2)$ is not complementary in the sense that $x^+ := x + d$ is not a node, since $x^+ = (1,1)$ and $Mx^+ + q = (0,5)$ are not complementary. Let us analyze this example a little further (see figure 2.1).

- The unique solution to the problem is the nonnegative vector $x^* = (2,0)$, since then $Mx^* + q = (0,4)$ is nonnegative and complementary.
- The B-Newton direction (1.5) is determined by the conditions

  $$d_1 + d_2 = 4 \qquad \text{and} \qquad 0 \leqslant (d_2 - 1) \perp (d_1 + 2d_2 - 1) \geqslant 0,$$

  whose unique solution is $d = (3,1)$, so that $x + d$ is the solution x^* to the problem.
- The polyhedra of the form $P_A := \{x : x_A \leqslant (Mx+q)_A,\ x_I \geqslant (Mx+q)_I\}$, with $I = [1\!:\!n] \setminus A$, are defined by $x_1 \leqslant x_1 + x_2 - 2$ and $x_2 \leqslant x_1 + 2x_2 + 2$, which after simplification become

  $$x_2 \geqslant 2 \qquad \text{and} \qquad x_1 + x_2 \geqslant -2.$$

  The nodes of the problem are $(0,0)$, $(0,-1)$, $x^* = (2,0)$, and $(6,-4)$. The displacement $d = (2,2)$ computed by (2.7) from $x = (-1,-1)$ goes through one of the nodes. □

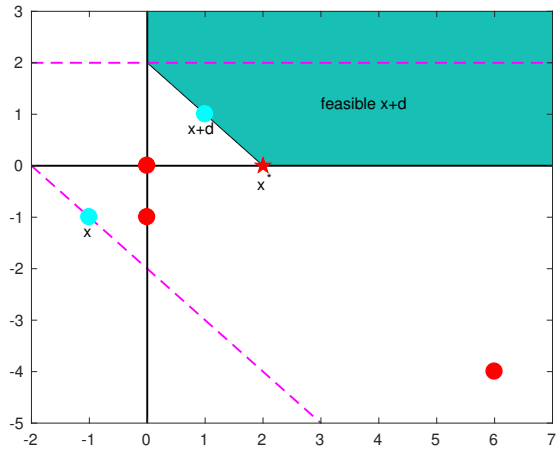

Figure 2.1: Counter-example 2.4. The solution x^* is the red star; the three other nodes of the problem are the red dots. The current point x and $x+d$ are the cyan dots, where d is computed by (2.9). The colored zone is $x+$"feasible set of (2.9)".

Since there is a direction d satisfying (2.7) and since this direction is a descent direction of Θ, it is natural to do a line search along it. If x^k is the current iterate and d^k is a solution to (2.7), the next iterate will be $x^{k+1} = x^k + \alpha^k d^k$, where the step size $\alpha^k > 0$ is computed by the Armijo rule: For some $\omega \in (0,1)$ independent of k, set $\alpha^k = 2^{-i^k}$, where i^k is the smallest nonnegative integer such that

$$\Theta(x^k + \alpha^k d^k) \leqslant (1 - 2\omega\alpha^k)\Theta(x^k). \tag{2.10}$$

Since $\Theta'(x^k; d^k) < 0$, it is clear that a positive step size $\alpha^k > 0$ can be determined by this rule.

2.2 Convergence

In this section, the global convergence of the Newton-min-convergent algorithm is established under slight assumptions. In addition to the previous assumptions, the convergence theorem only requires the problem to have at least one solution. Since the problem of determining whether a particular instance of a LCP has a solution is strongly NP-complete [10] and NP-complete for a $\mathbf{P}_0$-matrix (i.e., when M has nonegative principal minors) [24], it cannot be better than NP-complete for nonlinear complementarity problems. Consequently, in this paper, it will be assumed that problem (1.1) is solvable.

Assumption 3 *The complementarity problem 1.1 has at least one solution.*

If this assumption is not verified, the cluster point of Newton-min-convergent will be a nonzero local minimum of Θ.

Theorem 2.5 *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ semismooth and f such as assumptions 1 to 3 are verified. Then any cluster point $\bar{x}$ of $\{x^k\}$ the sequence generated by the Newton-min-convergent algorithm is a solution of (1.1).*

PROOF. Since Θ is a nonnegative function and by the Armijo rule (2.10) it is strictly decreasing, it follows that $\lim \alpha^k \Theta(x^k) = 0$. If $\liminf \alpha^k > 0$, then $\Theta(\bar{x}) = 0$.

The most difficult part is when $\liminf \alpha^k = 0$. The proof follows the analysis in [21] but see also [34]. Two intermediate results on which the analysis relies are

- $\Theta'(x, d) \leq -2\Theta(x^k) \leq 0$;
- $\exists d_\infty < \infty : \|d^k\| \leq d_\infty$, i.e. d^k is uniformly bounded.

The point $\hat{x}^k := x^k + 2\alpha^k d^k$ denotes the point just before the Armijo line search ended, which allows to write, denoting $\bar{\alpha}^k = 2\alpha^k$.

$$\Theta(\hat{x}^k) > \Theta(x^k) - 2\omega\bar{\alpha}^k\Theta(x^k). \tag{2.11}$$

The analysis consists in bounding below Θ term by term

$$\Theta(x^k) - \Theta(\hat{x}^k) = \sum(\min(x_i^k, f_i(x^k)))^2 - \sum(\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \tag{2.12}$$

as (to be proved below)

$$(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq 2\bar{\alpha}^k(\min(x_i^k, f_i(x^k)))^2 + o(\bar{\alpha}^k) \tag{2.13}$$

and then divide by $\bar{\alpha}^k$ and using also (2.11), sum to get

$$2\omega\Theta(x^k) \geq \frac{\Theta(x^k) - \Theta(x^k + \bar{\alpha}^k d^k)}{\bar{\alpha}^k} \geq 2\Theta(x^k) + \mathcal{O}(\bar{\alpha}^k) \tag{2.14}$$

and passing to the limit,

$$2\omega\Theta(\bar{x}) \geq 2\Theta(\bar{x}) \tag{2.15}$$

which implies $\Theta(\bar{x}) = 0$ since $\omega \in (0, 1)$.

Now, we prove (2.13). The points x^k and $\hat{x}^k$ are both close to $\bar{x}$ and we establish the bound (2.13) for all cases. In the following, the index set $E^0(\bar{x}) := \{i : x_i = f_i(x) = 0\}$ has to be considered separately.

- $i \in A(\bar{x})$: There exists $\tilde{k}_i$ large enough where $i \in A(x^k)\, \forall k \geqslant \tilde{k}$. Which means $(x_i^k)^2 - (\hat{x}_i^k)^2 = (x_i^k)^2 - (x_i^k + \bar{\alpha}^k d_i^k)^2 = -2x_i^k\bar{\alpha}^k d_i^k + o(\bar{\alpha}^k)$. Since $i \in A(x^k)$ and $d_i^k = -x_i^k$, it allows to conclude

$$(x_i^k)^2 - (\hat{x}_i^k)^2 = 2\bar{\alpha}^k(x_i^k)^2 + o(\bar{\alpha}^k) = 2\bar{\alpha}^k(\min(x_i^k, f_i(x^k)))^2 + o(\bar{\alpha}^k). \tag{2.16}$$

- $i \in I(\bar{x})$: There exists $\tilde{k}_i$ large enough where $i \in I(x^k)\, \forall k \geqslant \tilde{k}$. Which means $(f_i(x^k))^2 - (f_i(\hat{x}^k))^2 = (f_i(x^k))^2 - (f_i(x^k) + \bar{\alpha}^k\nabla f_i(x^k)d^k + \mathcal{O}(\|\bar{\alpha}^k d^k\|)^2)^2$. Since $i \in I(x^k)$ and $\nabla f_i(x^k)d^k = -f_i(x^k)$, it allows to conclude

$$(f_i(x^k))^2 - (f_i(\hat{x}^k))^2 = 2\bar{\alpha}^k(f_i(x^k))^2 + o(\bar{\alpha}^k) = 2\bar{\alpha}^k(\min(x_i^k, f_i(x^k)))^2 + o(\bar{\alpha}^k) \tag{2.17}$$

- $i \in (E^+(\bar{x})\backslash E^0)$: in this case, $x_i^k, f_i(x^k), \hat{x}_i^k, f_i(\hat{x}^k) > 0$ and $(\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 = \min((\hat{x}_i^k)^2, (f_i(\hat{x}^k))^2)$ so that $-(\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq -(\hat{x}_i^k)^2$ and $-(\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq -(\hat{f}_i(x^k))^2$. For this case, the proof is splitted according to the fact that the index i belongs to $A(x^k)$, $I(x^k)$ or $E(x^k)$.

– $i \in A(x^k)$: $\min(x_i^k, f_i(x^k)) = x_i^k$; there holds

$$(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq (x_i^k)^2 - (\hat{x}_i^k)^2. \tag{2.18}$$

Finally, (2.16) applies.

– $i \in I(x^k)$: $\min(x_i^k, f_i(x^k)) = f_i(x^k)$; there holds

$$(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq (f(x_i^k))^2 - (f(\hat{x}_i^k))^2 \tag{2.19}$$

and (2.17) applies.

– $i \in E(x^k)$: both cases above may apply.

- $i \in (E^-(\bar{x}) \backslash E^0)$: in this case, $x_i^k, f_i(x^k), \hat{x}_i^k, f_i(\hat{x}^k) < 0$ so that the index i belongs to $E^-(x^k) \cup A^-(x^k) \cup I^-(x^k)$. Also, $(\min(x_i^k, f_i(x^k)))^2 = \max((x_i^k)^2, (f_i(x^k))^2)$ so that $(\min(x_i^k, f_i(x^k)))^2 \geq (x_i^k)^2$ and $(\min(x_i^k, f_i(x^k)))^2 \geq (f_i(x^k))^2$. For this case, the proof is splitted according to the fact that the index i belongs to $A^-(\hat{x}^k)$, $I^-(\hat{x}^k)$ or $E^-(\hat{x}^k)$.

  – $i \in A(\hat{x}^k)$: $\min(\hat{x}_i^k, f_i(\hat{x}^k)) = \hat{x}_i^k$; we have

  $$(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq (x_i^k)^2 - (\hat{x}_i^k)^2. \tag{2.20}$$

  Since i is in $E^-(x^k) \cup A^-(x^k) \cup I^-(x^k)$ then we impose $d_i \geq -x_i^k$ and the reasoning above still applies to get (2.16).

  – $i \in I(\hat{x}^k)$: $\min(\hat{x}_i^k, f_i(\hat{x}^k)) = f_i(\hat{x}^k)$; we have

  $$(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2 \geq (f_i(x^k))^2 - (f_i(\hat{x}^k))^2. \tag{2.21}$$

  Since i is in $E^-(x^k) \cup A^-(x^k) \cup I^-(x^k)$ then we impose $\nabla f_i(x^k) d^k \geq -f_i(x^k)$ and the reasoning above still applies.

  – $i \in E(\hat{x}^k)$: both cases above may apply.

- $i \in E^0(\bar{x})$. In this case, rewrite $(\min(x_i^k, f_i(x^k)))^2 - (\min(\hat{x}_i^k, f_i(\hat{x}^k)))^2$ as the product of a difference factor and a sum factor

$$(\min(x_i^k, f_i(x^k))) - (\min(\hat{x}_i^k, f_i(\hat{x}^k))) \times (\min(x_i^k, f_i(x^k)) + (\min(\hat{x}_i^k, f_i(\hat{x}^k)))). \tag{2.22}$$

Since $\min(x, f(x))$ is Lipschitzian, the difference factor is $\mathcal{O}(x_i^k - \hat{x}_i^k) = \mathcal{O}(\bar{\alpha}^k)$ by the boundeness assumption on d^k. The sum factor converges to 0, thus the whole term is $o(\bar{\alpha}^k)$. Therefore, such terms, when divided by $\bar{\alpha}^k$, will vanish when taking the limit.

□

3 Newton-min-descent

In our experiments, the algorithm presented in section 2 frequently has to solve large quadratic problems which can increase the time taken by the algorithm to find the solution. To overcome this situation, Newton-min-descent algorithm is presented. This two phases algorithm is similar to Newton-min-convergent except it rarely solves quadratic problems and when it does the dimension is usually very low.

The first phase is the plain Newton method for nonsmooth equation applied to the min function (1.2). The calculation of d is performed by solving the equations

$$\begin{cases} (x+d)_{A\cup E_A} = 0 \\ (f(x) + \nabla f(x)d)_{I\cup E_I} = 0. \end{cases} \tag{3.1}$$

When E is nonempty, this direction is not unique because equations (3.1) depend of the partition of E. Since there is no information a priori on which choice of partition would be better, we chose $E_A = E$ and $E_I = \varnothing$ in our implementation in order to reduce the number of variables in the linear system to solve. It is known that this method is not necessarily convergent, even when f is linear and ∇f a $\mathbf{P}$-matrix [2, 5, 6]. Furthermore, it may generate directions along which Θ increases [3]. Actually, by the same reasoning of the proof of proposition 2.1 it is easy to show that with this direction the directional derivative of Θ along d is

$$\Theta'(x;d) = -2\Theta(x) + x_E^{\mathsf{T}} \min((x+d)_E, (f(x) + \nabla f(x)d)_E). \tag{3.2}$$

Hence, d will not be descending on Θ only if

$$x_{E^-}^{\mathsf{T}} \min((x+d)_{E^-}, (f(x)+\nabla f(x))_{E^-}) \geqslant 2\Theta - x_{E^+}^{\mathsf{T}} \min((x+d)_{E^+}, (f(x)+\nabla f(x))_{E^+}) \tag{3.3}$$

which is unlikely to occur frequently. It can still happen that is why (3.3) is tested to make sure d is descending and if not, the second phase is executed.

This second phase is another variation of the Newton method for nonsmooth equation applied to the min function (1.2). Like Newton-min-convergent it allows $F(x) + \hat{G}(x;d) \geqslant 0$ but only for the indices in the index set E^- which means d is the solution of the following optimization problem

$$\begin{cases} \min \frac{1}{2}\|d\|^2 \\ (x+d)_{A\cup E_A} = 0 \\ (f(x) + \nabla f(x)d)_{I\cup E_I} = 0 \\ (x+d)_{E^-} \geqslant 0 \\ (f(x) + \nabla f(x)d)_{E^-} \geqslant 0. \end{cases} \tag{3.4}$$

For the same reasons as above, in our implementation $E_A = E$ and $E_I = \varnothing$. Note that forcing $(x+d)_{E^-} \geqslant 0$ and $(f(x) + \nabla f(x)d)_{E^-} \geqslant 0$ leads to the same upper bound as before on Θ that is $\Theta'(x;d) \leqslant -2\Theta(x)$, hence d is descending. A line search respecting the Armijo rule (2.10) is then performed.

Since one of the most costly operation per iteration is solving a linear system and it is present in both phases, it is solved with the LU decomposition of $\nabla f_{II}(x)$ and it is stored in memory. Thus, if the phase 2 is needed most of the calculations for obtaining d_I are

done and the operations required for solving the quadratic subproblem will be the only ones that matter in terms of calculation time.

Unfortunately for this method, the step sizes along the Newton-min-descent direction may be arbitrarily small. Therefore, any theory ensuring the convergence when the step sizes are bounded away from zero cannot be applied for showing the convergence of any variant of this algorithm that coincides with it at all the points x with $E(x) = \varnothing$, and any alleged proof showing that the step sizes are bounded away from zero must contain some mistake[1].

Counter-example 3.1 (step sizes not bounded away from zero) Consider the Murty linear complementarity problem in dimension 2 (see section 5.1), a starting point x_t parameterized by $t > 0$ and $f(x) = Mx + q$ where

$$n = 2, \qquad M = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, \qquad q = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \qquad \text{and} \qquad x_t = \begin{pmatrix} 1/2 - t \\ -1/2 \end{pmatrix}. \tag{3.5}$$

Note that M is a $\mathbf{P}$-matrix. Since $Mx_t + q = (\frac{1}{2} - t, -\frac{1}{2} - 2t)^\mathsf{T}$, one gets $A(x_t) = \varnothing$, $I(x_t) = \{1, 2\}$, and $E(x_t) = \varnothing$, so that F is differentiable at x_t and the direction d^{NM} respecting (3.1) is uniquely determined. Since $F(x_t) = Mx_t + q$, it reads

$$d_t^{\text{NM}} = \begin{pmatrix} \frac{1}{2} + t \\ -\frac{1}{2} \end{pmatrix}.$$

The map F has for unique kink the vertical line $\{1/2\} \times \mathbb{R}$. The map $\alpha \in [0, 1] \mapsto \Theta(x_t + \alpha d_t^{\text{NM}})$ is quadratic decreasing on $[0, \alpha_t]$ where $\alpha_t := 2t/(1 + 2t)$ brings $x_t + \alpha_t d^{\text{NM}}$ on the kink of F. While on the other segment, $[\alpha_t, \infty)$ it continue to decrease until

$$\alpha_t^{\min} := \frac{3t^2 + 2t}{3t^2 + 2t + \frac{1}{2}} > \alpha_t$$

and is quadratic increasing thereafter. To see this, observe that the point $\check{x}_t = x_t + \alpha_t d_t^{\text{NM}} = (\frac{1}{2}, -(\frac{1}{2} + 2t)/(1 + 2t))^\mathsf{T}$ is such as

$$\Theta'(\check{x}_t; -d_t^{\text{NM}}) = \frac{\frac{1}{2} + 3t + 5t^2}{1 + 2t} > \frac{1}{2}$$
$$\Theta'(\check{x}_t; d_t^{\text{NM}}) = -\frac{t^2}{1 + 2t} < 0,$$

so that there is a jump in the directional derivatives. A laborious calculation, then shows that $\Theta(x_t + \alpha d^{\text{NM}}) \leqslant \Theta(x_t)$ if and only if $\alpha \in [0, \bar{\alpha}_t]$, where

$$\bar{\alpha}_t := \frac{t(1 + t) + \sqrt{t(1 + 5t + 8t^2 + 5t^3)}}{\frac{1}{2} + t + t^2}.$$

Since $\bar{\alpha}_t \to 0$ when $t \downarrow 0$, it follows that a step size ensuring the decrease of Θ may be arbitrary small. □

[1] [1; theorems 3.1 and 3.2]

As we shall see in section 5, Newton-min-descent may not be convergent. During our experiments, we encountered a situation where the sequence of step sizes decrease slowly under the unit roundoff. However it is still worth using its direction since it is not expensive to compute. In the next section, an hybrid algorithm is presented to keep both the global convergence and the better results in most applications.

4 Newton-min-hybrid algorithm

The Newton-min-hybrid algorithm uses all directions presented in sections 2 and 3. The main idea is to compute the Newton-min-convergent direction only as a last resort since its subproblem is generally the hardest to solve.

If we suppose that $\liminf \alpha^k > 0$ Newton-min-descent is convergent. Moreover Θ is also strictly decreasing, so the only reason an cluster point $\bar{x}$ of would not be a solution of (1.1) is if $\liminf \alpha^k = 0$ and $\Theta(\bar{x}) \neq 0$. An easy way to avoid this scenario is to always compute the Newton-min-descent direction unless the step sizes become too small. Given $f(x)$ and $\nabla f(x)$ the Newton-min-hybrid algorithm is defined as follow:

initialization : starting point $x \in \mathbb{R}^n$, $\eta \in (0,1)$, $\omega \in (0, \frac{1}{2})$, termination criterion $\varepsilon > 0$, previous step sizes memory $\mu \in \mathbb{N}$, threshold on step sizes $\tau \in (0,1]$, previous iteration memory $\gamma \geqslant 2$;
while $\Theta(x) > \varepsilon$ **do**
  if *one of the* μ *previous step sizes* $> \tau$
  or all previous γ *iterations are Newton-min-convergent* **then**
    Compute the first phase descent direction (3.1);
    if $x_E^t \min((x+d)_E, (f(x)+\nabla f(x))_E) \geqslant 2\eta\Theta$ **then**
      Compute the second phase descent direction ($\min d$ respecting (3.4));
    end
  else
    Compute Newton-min-convergent direction;
  end
  Compute the step size α with the armijo rule (2.10);
  $x \leftarrow x + \alpha d$;
end

Algorithm 1: Newton-min-hybrid algorithm.

Theorem 4.1 *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ semismooth and f such as assumptions 1 to 3 are verified. Then any cluster point $\bar{x}$ of $\{x^k\}$ the sequence generated by the Newton-min-hybrid algorithm is a solution of (1.1).*

Proof. If $\liminf \alpha^k > 0$, it follows that $\Theta(\bar{x}) = 0$ for the same reason as theorem 2.5 because both Newton-min-convergent and Newton-min-descent directions ensure $\Theta(x^{k+1}) < \Theta(x^k)$. The case where $\liminf \alpha^k = 0$ is trivial since there will be a sufficiently large value $\tilde{k}$ where all the step sizes α^k will be lower than τ for $k \geqslant \tilde{k}$. Then from iteration $\tilde{k} + \mu$ the sequence of directions computed will be $\gamma - 1$ Newton-min-convergent and one Newton-min-descent alternately. □

4.1 Implementation

To reduce the size of the quadratic problem to solve, in our implementation the constraints are expressed only in terms of d_{E^-} (or $d_{E^-\cup A^-\cup I^-}$). To lighten the text, we consider only the quadratic subproblem of Newton-min-descent. An easy computation provides

$$\begin{aligned} d_A &= -x_A, \\ d_I &= -\nabla f_{II}^{-1}(x)\nabla f_{IE^-}(x)d_{E^-} - v_I, \end{aligned}$$

where we have introduced the vector

$$v_I := \nabla f_{II}^{-1}(x)\left(f_I(x) - \nabla f_{IA}(x)x_A\right).$$

Then the inequality constraints that d_{E^-} must verify are

$$\begin{aligned} d_{E^-} &\geqslant -x_{E^-}, \\ \left(\nabla f_{E^-E^-}(x) - \nabla f_{E^-I}(x)\nabla f_{II}^{-1}(x)\nabla f_{IE^-}(x)\right) d_{E^-} &\geqslant -f_{E^-}(x) + \nabla f_{E^-A}(x)x_A + \nabla f_{E^-I}(x)v_I. \end{aligned}$$

Ignoring the constant component $d_A = -x_A$, the objective $\frac{1}{2}\|d\|^2$ can be written, hence up to an additive constant:

$$\begin{aligned} \frac{1}{2}\left\|\begin{pmatrix}\nabla f_{II}^{-1}(x)\nabla f_{IE^-}(x)d_{E^-} + v_I \\ d_{E^-}\end{pmatrix}\right\|^2 = \frac{1}{2}d_{E^-}^{\mathsf{T}}\left(\nabla f_{IE^-}(x)^{\mathsf{T}}\nabla f_{II}^{-1}(x)^{\mathsf{T}}\nabla f_{II}^{-1}(x)\nabla f_{IE^-}(x) + I\right) d_{E^-} \\ + v_I^{\mathsf{T}}\nabla f_{II}^{-1}(x)\nabla f_{IE^-}(x)d_{E^-} + \frac{1}{2}\|v_I\|^2. \end{aligned}$$

4.2 Linear complementarity problems

As mentioned before, the strength of the algorithms presented is that they are applicable when f is linear. In this section, let suppose $f = Mx + q$ where $M \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ thus $\nabla f(x) = M$. Our claim is that the Newton-min-hybrid reach the solution of a linear complementarity problem in a finite number of iterations. It comes from the fact that the first phase descent direction compute the global minimum of the quadratic piece of Θ at x and not an approximation since $f(x + \alpha d) = Mx + q + Md$.

Proposition 4.2 *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be defined as $f(x) = Mx+q$ with $M \in \boldsymbol{S}$ nondegenerate and assumption 3 verified. Then the Newton-min-hybrid algorithm find a solution of (1.1) in a finite number of iterations.*

Proof. Let x^* be a solution Newton-min-hybrid tends to. By theorem 4.1, there exists $\tilde{k}$ such that $A(x^k) \supseteq A(x^*)$ and $I(x^k) \supseteq I(x^*)$ for all $k \geqslant \tilde{k}$. Moreover, by the definition of the algorithm it is certain that phase one descent direction iteration will be done for $x^{\hat{k}}$ where $\hat{k} \geqslant \tilde{k}$. If $E(x^*) = \varnothing$, it is obvious that $x^{\hat{k}+1} = x^*$. If not, by definition it follows

$$0 = x^{\hat{k}+1}_{A(x^*)} = x^*_{A(x^*)} \qquad \text{and} \qquad 0 = (Mx^{\hat{k}+1} + q)_{I(x^*)} = (Mx^* + q)_{I(x^*)}.$$

For the remaining indices, since $x^*_i = (Mx^* + q)_i = 0$ the choice of E_A and E_I has no impact because the first phase descent iteration nullifies $x_i^{\hat{k}+1}$ or $(Mx^{\hat{k}+1} + q)_i$. □

Note that the greatest cost per iteration of Newton-min-hybrid applied to LCP is only two matrix vector products, that is, Mx and Md, the resolution of a linear system of dimension $\mathbf{card}(I)$ (or $\mathbf{card}(I^+)$) and perhaps solving a quadratic problem of lower dimension.

5 Numerical experiments on LCPs

First, Newton-min-hybrid is tested on Murty and Fathi problems which are known to be difficult. Then it is compared with other methods on random problems generator and problems obtained from applications. The algorithms are also tested on a scaled version of the problem which is expected to improve their performance when f is linear.

The scaling used $S2$, divides each lines of q and M by its norm, namely $M \leftarrow S_2 M;\ q \leftarrow S_2 q$ where

$$S_2 = \begin{pmatrix} \frac{1}{||l_1||} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{||l_2||} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{||l_3||} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{||l_n||} \end{pmatrix},$$

with $l_1, l_2, \ldots, l_n$ the lines of M. This scaling does not affect the solution. Indeed it is represented by a positive diagonal matrix multiplication hence the solution of $0 \leqslant x \perp S_2(Mx+q) \geqslant 0$ is the same as the original problem.

The followings tests are performed on a machine with a 8 cores intel i7-4771 processor and the memory available is 32 GB. The algorithm is implemented in Matlab and the quadratics subproblems are solved with quadprog solver. The parameters used are $\omega = 10^{-4}$, $\eta = \frac{7}{8}$, $\mu = 4$, $\tau = 0.1$ and $\gamma = 10$ for Newton-min-hybrid. The calculation time (in seconds) includes all the time needed to the resolution of the problem. The column "Qps" indicates how many quadratic subproblems Newton-min-hybrid had to solve and "LargestQp" is the one of largest dimension. Finally, the column "M*v" indicates the number of matrix-vector multiplication performed since it may vary from iteration to another for Minmap-Newton.

5.1 Murty and Fathi problems

The Murty problem [31] is often considered to have the following data M and q, and starting point x:

$$M = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 2 & 1 & 0 & \cdots & 0 \\ 2 & 2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 2 & 2 & \cdots & 1 \end{pmatrix}, \qquad q = -\mathbf{1}, \qquad \text{and} \qquad x = 0,$$

where $\mathbf{1}$ is the vector of all ones, although other values of q are considered in [32; chapter 6]. The matrix M is clearly a $\mathbf{P}$-matrix (its principal minors have the value 1). This problem is extensively used [29, 22, 9], probably because some pivoting methods [30] are known to

	No scaling				Scaling S2			
n	iter	Qps	LargestQP	time	iter	Qps	LargestQP	time
512	182	116	293	0.804	6	1	512	0.056
2048	378	295	1259	38.0	6	1	2048	1.31
8192	1538	1362	5082	3099	6	1	8192	42.1
16384	Stopped after 20000 seconds				6	1	16384	400

Table 5.1: Results on the Murty problem

	No scaling				Scaling S2			
n	iter	Qps	LargestQP	time	iter	Qps	LargestQP	time
512	70	1	10	0.095	6	1	512	0.848
2048	67	1	1978	2.40	6	1	2048	38.3
8192	70	1	8125	848	6	1	8192	2452
16384	69	1	16314	10223	6	1	16384	9795

Table 5.2: Results on the Fathi problem

require an exponential number of iterations to solve it [32; theorem 6.4]. This problem is also difficult to solve for the Newton-min-hybrid algorithm but not with the same severity (note also that the Newton-min-hybrid algorithm finds the solution in one iteration from $x = \mathbf{1}$).

In the Fathi problem [19; 1979], M, q, and the starting point are given by

$$M = \begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 2 & 5 & 6 & \cdots & 6 \\ 2 & 6 & 9 & \cdots & 10 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 6 & 10 & \cdots & 4(n-1)+1 \end{pmatrix}, \qquad q = -\mathbf{1}, \qquad \text{and} \qquad x = 0.$$

Since $M = M_1 M_1^{\mathsf{T}}$, where M_1 is the Murty matrix, and since $M_1 \in \mathbf{P}$, M is symmetric positive definite, hence a $\mathbf{P}$-matrix.

In both of these problems when the number of variables is large enough the algorithm generates directions such as the step sizes become too small so convergent iteration are needed. Furthermore, the scaled problem is such as all the indices are in $A^- \cup I^- \cup E^-$ after five iterations so a Qp of n variables has to be solved and leads to the solution in one step.

5.2 Comparison with PATH and Minmap-Newton on random problems

The PATH solver for mixed complementarity problems (MCPs) was introduced in 1995 by Dirske and Ferris [15] and has since become the standard against which new MCP solvers are compared[7, 25]. It is an implementation of a stabilized Newton method for the solution of the MPC. The stabilization scheme employs a path-generation procedure which is used to construct a piecewise-linear path from the current point to the Newton point. The core algorithm is to find a zero of the normal map

$$F_B(x) := F(\pi_B(x)) + x - \pi_B(x)$$

	No scaling			Scaling S2		
	Newton-min-hybrid					
n	iter	Qps	time	iter	Qps	time
4096	6	0	0.934	6	0	0.939
8192	6	0	3.43	6	0	3.38
16384	6	0	21.4	7	0	26.0
32768	6	0	163	7	0	182
	Minmap-Newton					
n	iter	M*v	time	iter	M*v	time
4096	70	430	95.2	7	13	7.49
8192	120	838	801	7	13	41.8
16384	150	1014	4908	7	13	252
32768	Out of memory			Out of memory		
	PATHLCP					
n	time			time		
4096	24.6			24.7		
8192	135			134		
16384	1723			1026		
32768	Out of memory			Out of memory		

Table 5.3: Results H-P generator

associated with the MCP where π_B represents the Euclidean projection of x into the bounds constraints B. If x^* is a zero of the normal map, then it solves the MCP. The solver has been fine-tuned over time and Pathlcp is an optimized version for LCPs. The comparison of Newton-min-hybrid and Pathlcp can only be done on the computing time, since the iterations of the two algorithms do not count for the same number of operations. Pathlcp is spectacularly efficient on the Murty problem and always dramatically outperforms Newton-min-hybrid.

The Minmap-Newton algorithm computes the first phase descent direction by solving (3.1) and performs a projected line search on $x \geqslant 0$ along it [17]. The cost per iteration is low and similar to the Newton-min-descent algorithm. It shows very good results on some applications but it can be shown that this direction is not always descending for Θ which makes it harder to prove its convergence. Indeed the author mentions the algorithm fails for some kind of problems.

5.2.1 Harker and Pang problem generator

The problem generator presented in [22] consist of LCPs with matrices M computed as followed. An $n \times n$ matrix A is randomly generated with uniformly distributed entries in $(-5, 5)$ and a skew-symmetric matrix is generated in the same interval. The matrix M is then defined by

$$M = A^T A + B + \text{Diag}(\epsilon_1, \ldots, \epsilon_n),$$

with $\epsilon \in (0, 0.3)$ while the vector q is generated from a uniform distribution in the interval (-500, 500).

	No scaling			Scaling S2		
	Newton-min-hybrid					
n	iter	Qps	time	iter	Qps	time
4096	5	0	0.900	1	0	0.089
8192	5	0	3.71	1	0	0.427
16384	5	0	18.0	2	0	4.08
32768	6	0	134	1	0	13.2
	Minmap-Newton					
n	iter	M*v	time	iter	M*v	time
4096	6	59	1.73	2	3	0.366
8192	6	59	9.36	2	3	2.18
16384	25	966	340	2	3	13.8
32768	6	59	302	2	3	74.9
	PATHLCP					
n	time			time		
4096	Failed			0.050		
8192	Failed			0.248		
16384	Failed			1.43		
32768	Failed			7.31		

Table 5.4: Results on the modified Harker and Pang sparse random problem generator with $L = 10^{15}$

Table 5.3 shows that these random problems are easy to solve for Newton-min-hybrid. The number of iteration is quasi-constant and the scaling does not improve its performance while it does for Minmap-Newton and Pathlcp.

5.2.2 Modified Harker and Pang problem generator for hard problems

Since the Harker and Pang problem generator creates easy problems for Newton-min-hybrid, we modified it in several ways to improve its difficulty. First of all, we forced the solution x^* not to be strictly complementary by choosing the vector q such as $\mathbf{card}(E(x^*)) = \frac{n}{5}$ and $\mathbf{card}(A(x^*)) = \mathbf{card}(I(x^*)) = \frac{2n}{5}$. Then we chose to create only sparse matrices. The reason why is the code available online for Pathlcp is only implemented for such matrices. Hence the comparisons can be made on an equal footing. The density of M in the two following tests is in between 2.49% and 2.62%.

Two additional modifications of M are considered. The first one is to take

$$M = A^T A + B + \mathrm{Diag}(v),$$

where v is a vector with positive values uniformly distributed between 0 to L. This has the effect of raising the condition number of M for large value of L.

The first thing we noticed on this set of LCPs is without any scaling Pathlcp fails to solve these problems as soon as L gets higher than 10^{10}. Furthermore, Newton-min-hybrid still has the best performance for large values of n but this lead does not hold on the scaled problem. Table 5.4 shows it may well solve LCP with very ill-conditioned. However those three algorithms all fail to solve the problems when L gets larger.

	No scaling			Scaling S2		
	Newton-min-hybrid					
n	iter	Qps	time	iter	Qps	time
4096	74	0	13.9	19	0	2.42
8192	33	0	21.1	18	0	9.17
16384	35	0	137	18	0	50.4
32768	27	0	718	17	0	353
	Minmap-Newton					
n	iter	M*v	time	iter	M*v	time
4096	Failed		> 20000	21	48	6.73
8192	Failed		> 20000	18	42	32.3
16384	Failed		> 20000	20	47	212
32768	Failed		> 20000	25	52	1626
	PATHLCP					
n	time			time		
4096	30.5			28.0		
8192	328			289		
16384	2773			3011		
32768	19181			18621		

Table 5.5: Results on the modified Harker and Pang random problem generator sparse with high l_0

The goal of the second modification is to raise the quantity

$$l_0(M) = \min_{||x||_\infty=1} \max_{1\leqslant i\leqslant n} x_i(Mx)_i$$

which was introduced by Mathias and Pang [27]. They have demonstrated this value can be used to derive error bounds for the linear complementarity problem. A simple way to raise it is to consider a large constant s where

$$M = A^T A + sB + \text{Diag}(\varepsilon_1, \ldots, \varepsilon_n).$$

This last modification makes the problems much harder for all algorithms as we see in table 5.5. On the unscaled problem, after only a few iterations the progress of Minmap-Newton seems to stops and all subsequent iterations are very close. This may indicate it does not converge to a solution but it finds one on the scaled problem. Newton-min-hybrid still has the best overall performance due to the slower increase of calculation. Another observation is that the scaling affect both Newton-min-hybrid and Minmap-Newton in a positive way but Pathlcp results are about the same on both versions of the problem.

5.2.3 Physics problem generator

One can find applications in solving LCPs in physics. The 3D fluid motion problem may be represented by a LCP with M a banded matrix with three off-diagonal bands. For our experiments the Matlab code of Erleben [18] is used to generate M then a solution x^*

	No scaling			Scaling S2		
	Newton-min-hybrid					
n	iter	Qps	time	iter	Qps	time
1000000	3	0	1.05	3	0	1.03
8000000	3	0	9.53	3	0	9.40
27000000	3	0	32.8	3	0	32.2
	Minmap-Newton					
n	iter	M*v	time	iter	M*v	time
1000000	4	7	1.04	4	7	1.00
8000000	4	7	8.59	4	7	8.39
27000000	4	7	29.0	4	7	28.7
	PATHLCP					
n	time			time		
1000000	5.00			5.24		
8000000	43.8			46.6		
27000000	178			174		

Table 5.6: Results on the fluid LCPs.

is randomly generated with all components in $[0, 1]$. Secondly, for all $x_i^* < \frac{1}{4}$ we choose $x_i^* = 0$ and then q is computed to be consistent with the solution. Results in table 5.6 are shown for a few values of n. More tests were executed for values in between and they showed the progression in calculation time was about linear and the number of iterations quasi-constant for Newton-min-hybrid and Minmap-Newton. An interesting observation in table 5.6 is that the scaling does not improve the performance of any algorithms. Actually the LCPs are so easy and the number of variables so large that scaling M takes about four times more time than Newton-min-hybrid takes to solve it. Here, the performance of Newton-min-hybrid and Minmap-Newton are similar and outperform Pathlcp.

Contact simulation problems provide another application where we solve LCPs. The contact coefficient matrix is filled with block matrices. The average percentage fill-in we used is 5% so we obtain sparse matrices again and can compare with Pathlcp fairly. Since these matrices are only $\mathbf{P}_0$-matrices, Newton-min-hybrid and Minmap-Newton algorithms are not well defined. Given a current iterate, it is possible that the linear system to solve $(M(x+d)+q)_I = 0$ has no solution. That is why an approximation of M is used to solve the problem. The algorithms are given $M + \epsilon I$ what may change the solution but not in a significant way. In table 5.7, the solution and vector q are made the same way as with fluid LCPs. The value of ϵ used in the tests is 10^{-15}.

For this kind of problem, one can see that Newton-min-hybrid may not be the fastest algorithm to find the solution for small values of n, but table 5.7 shows that it is more robust than Minmap-Newton and Pathlcp. Again, Newton-min-hybrid has a slower increase of the calculation time compared to Pathlcp. Moreover with a simple perturbation Newton-min-hybrid still finds the solution of the LCP even if M is degenerate.

An interesting observation made during our experiments is that for larger contact problems an iteration of Newton-min-convergent has to be made. Indeed, if only Newton-min-descent is used to solve these problems with 12000 and 18000 variables, the second

	No scaling			Scaling S2		
	Newton-min-hybrid					
n	iter	Qps	time	iter	Qps	time
3000	2	0	0.325	2	0	0.415
6000	2	0	1.27	2	0	1.50
12000	15	1	314	10	1	268
18000	16	1	1125	10	1	1083
	Minmap-Newton					
n	iter	M*v	time	iter	M*v	time
3000	3	5	0.115	3	5	0.118
6000	3	5	1.35	3	5	1.34
12000	Failed			Failed		
18000	Failed			Failed		
	PATHLCP					
n	time			time		
3000	0.105			0.089		
6000	24.9			9.90		
12000	2070			632		
18000	Failed			Failed		

Table 5.7: Results on the contact LCPs. Minmap-Newton fails because the submatrix is close to be singular.

step size is around 10^{-10} and the sequence $\{\alpha^k\}$ decreases around 10^{-20} before moving up again around 10^{-11} and decreasing again and so on until it finds a solution or it fails. This may indicate that Newton-min-descent is not convergent for this kind of problem but if a rounding error changes the index sets it could ensure the convergence. Hence it justifies the need of the convergent iterations.

5.3 Observations

During our experiments, we noticed that the scaling S2 may be appropriate on harder problems. When the problems are made such as an index of difficulty is relatively high or when Newton-min-hybrid takes more than about ten iterations, the scaling always improves the results. Actually in all our experiments it has never returned a significantly more difficult problem. With our implementation, the scaling takes about 4 milliseconds when M is a full 512×512 matrix and about 1 second when M is a full 8192×8192 matrix. Hence it might enhance the performance of Newton-min-hybrid on smaller instance to consider S2.

Another observation made during our experiments is that Newton-min-hybrid must perform a line search. On easier problems like the Harker and Pang generator, the unit step size is usually taken for all iterations. Moreover, on most applications the unit step size is also mainly taken and Newton-min-descent still finds the solution if we force a unit step size. In this case, the sequence $\{\Theta(x^k)\}$ may not be monotone decreasing but it does not seem to affect significantly the number of iteration to find a solution on easy instances. On the other hand, on harder problems such as our modified H-P generator the line search is essential. If we force the algorithm to take the next iterate $x + d$ on these problem

Newton-min-hybrid may cycle. Indeed, for every instance tested the progress on Θ appears to stop after a few iterations and $\Theta(x^k)$ stabilize away from zero. For example, on the problems ill-conditioned, after 500 iterations Newton-min-hybrid without line search has not yet found the solution on every instances while it never takes more than six with a line search in table 5.4. Note that the same observation can be made on the problem scaled by S2.

6 Conclusion

This paper presents a modification of the semismooth Newton method with a guaranteed descent of the minimum least-square merit function. It is shown that the Newton-min-hybrid algorithm is globally convergent and well-defined under some assumptions. No LCP subproblem has to be solve. This is an important property which is not possessed by any similar globalization approach. The results obtained numerically on a large range of problems show that the hybrid algorithm is dramatically efficient on most application and is competitive with state-of-the-art solvers.

In future works, it would be interesting to investigate the properties of a similar algorithm with a different choice of descent direction. The minimal ℓ_1 or ℓ_∞ norms could be considered. Indeed, computing the minimal ℓ_1 norm would lead to a linear optimization subproblem. Thereafter, a complexity analysis of the algorithm could be made. It might also be interesting to improve the efficiency of the quadratic subproblem solver by solving it on GPU for instance.

References

[1] Z.-Z. Bai, J.-L. Dong (2006). A modified damped Newton method for linear complementarity problem. *Numerical Algorithms*, 42, 207–228. [doi]. 49

[2] I. Ben Gharbia, J.Ch. Gilbert (2012). Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134, 349–364. [doi]. 48

[3] I. Ben Gharbia, J.Ch. Gilbert (2012). A simple example of semismooth Newton direction that is not a descent direction of the least-square merit function. Working notes. 48

[4] I. Ben Gharbia, J.Ch. Gilbert (2012). On the Newton-min algorithm for solving linear complementarity problems. Working notes. 40

[5] I. Ben Gharbia, J.Ch. Gilbert (2013). An algorithmic characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3), 904–916. [doi]. 48

[6] I. Ben Gharbia, J.Ch. Gilbert (2018). An algorithmic characterization of P-matricity II: corrections, refinements, and validation. *SIAM Journal on Matrix Analysis and Applications* (in revision). [preprint]. 48

[7] S.C. Billups, S.P. Dirkse, M.C. Ferris (1997). A comparison of large scale mixed complementarity problem solvers. *Computational Optimization and Applications*, 7, 3–25. 53

[8] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (2006). *Numerical Optimization – Theoretical and Practical Aspects* (second edition). Universitext. Springer Verlag, Berlin. [authors] [editor] [doi]. 41, 42

[9] B. Chen, P.T. Harker (1993). A non-interior continuation method for linear complementarity problems. *SIAM Journal on Matrix Analysis and Applications*, 14, 1168–1190. [doi]. 52

[10] S.J. Chung (1989). NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60, 393–399. [doi]. 45

[11] F.H. Clarke (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York. 40

[12] A.R. Conn, N.I.M. Gould, Ph.L. Toint (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization 1. SIAM and MPS, Philadelphia. [doi]. 41

[13] R.W. Cottle, J.-S. Pang, R.E. Stone (1992). *The Linear Complementarity Problem*. Academic Press, Boston. 43

[14] J.E. Dennis, R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs. 41

[15] S.P. Dirkse, M.C. Ferris (1995). The path solver: a nommonotone stabilization scheme for mixed complementarity problems. *Optimization Methods and Software*, 5(2), 123–156. 41, 53

[16] J.-P. Dussault, M. Frappier, J.Ch. Gilbert (2018). A lower bound on the iterative complexity of the Harker and Pang globalization technique of the Newton-min algorithm for solving the linear complementarity problem. *EURO Journal on Computational Optimization* (submitted). [hal-01806399].

[17] K. Erleben (2013, Febuary). Numerical methods for linear complementarity problems in physics-based animation. Siggraph course notes. 54

[18] K. Erleben, M. Andersen, S. Niebe, M. Silcowitz (2011, October). NUM4LCP. Published online at http://github.com/erleben/num4lcp. Open source project for numerical methods for linear complementarity problems in physics-based animation. 56

[19] Y. Fathi (1979). Computational complexity of LCPs associated with positive definite symmetric matrices. *Mathematical Programming*, 17, 335–344. 53

[20] J.Y. Han, D.F. Sun (1997). Newton and quasi-Newton methods for normal maps with polyhedral sets. *Journal of Optimization Theory and Applications*, 94, 659–676. [doi]. 40

[21] S.-P Han, J.-S. Pang, N. Rangaraj (1992). Convergent Newton Methods for Nonsmooth Equation. *Mathematics of Operations Research*, 17, 586–607. [doi]. 41, 46

[22] P.T. Harker, J.-S. Pang (1990). A damped-Newton method for the linear complementarity problem. In E.L. Allgower, K. Georg (editors), *Computational Solution of Nonlinear Systems of Equations*, Lecture in Applied Mathematics 26. AMS, Providence, RI. 52, 54

[23] A.F. Izmailov, M.V. Solodov (2014). *Newton-Type Methods for Optimization and Variational Problems*. Springer Series in Operations Research and Financial Engineering. Springer. 41

[24] M. Kojima, N. Megiddo, T. Noma, A. Yoshise (1991). *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems*. Lecture Notes in Computer Science 538. Springer, Berlin. [doi]. 45

[25] Q. Li, M.C. Ferris, T. Munson (2009). Linear algebra enhancements to the path solver. Technical report, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, Illinois, USA. 53

[26] X. Li, D. Sun, K.-C. Toh (2017). On the efficient computation of a generalized Jacobian of the projector over the Birkhoff polytope. Technical report. arXiv:1702.05934. 40

[27] R. Mathias, J.-S. Pang (1990). Error bounds for the linear complementarity problem with a P-matrix. *Linear Algebra and its Applications*, 132, 123–136. 56

[28] R. Mifflin (1977). Semismooth and semiconvex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15(6), 959–972. [doi]. 40

[29] S. Mizuno, A. Yoshise, T. Kikuchi (1989). Practical polynomial time algorithms for linear complmentarity problems. *Journal of Operations Research Society of Japan*, 32(1), 75–92. [doi]. 52

[30] K.G. Murty (1974). Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, 11, 123–130. 52

[31] K.G. Murty (1978). Computational complexity of complementarity pivot methods. *Mathematical Programming Study*, 7, 61–73. 52

[32] K.G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming* (Internet edition, prepared by Feng-Tien Yu, 1997). Heldermann Verlag, Berlin. 44, 52, 53

[33] J.-S. Pang (1990). Newton's method for B-differentiable equations. *Mathematics of Operations Research*, 15, 311–341. [doi]. 40

[34] J.-S. Pang (1991). A b-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational in-equality problems. *Mathematical Programming*, 51, 101–131. 40, 46

[35] L. Qi, J. Sun (1993). A nonsmooth version of Newton's method. *Mathematical Programming*, 58, 353–367. [doi]. 40

[36] D. Ralph (1994). Global convergence of damped Newton's method for nonsmooth equations via the Path search. *Mathematics of Operations Research*, 19(2), 257–512. [doi]. 41

[37] M.V. Solodov, B.F. Svaiter (2000). A truly globally convergent Newton-type method for the monotone nonlinear complementarity problem. *SIAM Journal on Optimization*, 10(2), 605–625. [doi]. 41

CONCLUSION

L'objectif initial était de travailler sur les problèmes de complémentarité et de trouver une globalisation à la méthode de Newton appliquée au cas linéaire défini par une **P**-matrice. Dans cet ouvrage, nous présentons une nouvelle méthode qui résout le problème de complémentarité pour un plus large éventail de fonctions. La preuve de convergence développée étant plus générale nous permet d'appliquer l'algorithme au cas non-linéaire. De plus, la méthode n'a pas besoin de résoudre de problème de complémentarité linéaire dans le calcul d'une direction ce qui constitue une innovation importante en comparaison des variantes de la méthode de Newton dans la littérature. L'implémentation est toutefois optimisée pour les problèmes de complémentarité linéaire et n'est applicable seulement que pour ceux-ci présentement. La prochaine étape serait donc d'adapter l'implémentation afin que Newton-min-hybrid puisse résoudre des problèmes de complémentarité non-linéaire. Il serait aussi intéressant de lui permettre de résoudre le problème généralisé suivant

$$\begin{gathered} 0 \leq f(x) \perp g(x) \geq 0 \\ l \leq x \leq u. \end{gathered}$$

Avec cette implémentation, il serait possible de comparer notre algorithme avec d'autres méthodes de pointe sur une plus grande variété de problèmes. D'autres avenues restent possibles aussi quant au choix de la direction de descente choisie. En effet, la preuve de convergence présentée s'applique pour toutes les directions qui satisfont aux équations (2.2) de l'article présenté au chapitre 3 tant qu'elles restent bornées. Un choix a du

être fait et nous avons opté pour la direction de norme ℓ_2 minimale parce qu'elle est unique et assez simple à calculer. Cependant, parmi les autres candidats possibles, il serait intéressant de tester une direction de norme ℓ_1 minimale car cela n'entraîne que la résolution d'un problème d'optimisation linéaire. Ce choix n'a pas été adopté initialement car le résultat peut ne pas être unique. Pour terminer, il serait remarquable de pouvoir démontrer si le problème de complémentarité linéaire définit par une **P**-matrice se résout en temps polynomial ou non.

Afin d'aboutir à une globalisation de Newton-min, un autre objectif de la maîtrise était de passer en revue diverses méthodes pour arriver à mieux comprendre la problématique. Lors de cette revue, j'ai été amené à implémenter l'algorithme de Harker et Pang ce qui a conduit au résultat du chapitre 2 sur la complexité itérative de cette méthode. Ce résultat garantit qu'en pire cas l'algorithme de Harker et Pang prendra au moins n itérations à trouver la solution où n est le nombre de variables. Le problème de Fathi a été créé spécifiquement pour donner de la difficulté à certains algorithmes. En effet, nos résultats numériques présentés dans l'article du chapitre 3 montrent qu'il s'agit d'un problème plutôt difficile à résoudre pour Newton-min-hybrid. Cependant, le résultat présenté sur la complexité itérative ne contredit pas l'hypothèse qu'une globalisation de Newton-min pourrait résoudre le problème de complémentarité linéaire définit par une **P**-matrice en temps polynomial ce qui laisse la porte ouverte à cette possibilité.

ANNEXE A

Dans la présente annexe, nous faisons la démonstration détaillée que la preuve apportée par Bai et Dong [BD06] prétendant que leur variante de l'algorithme de Newton-min avec recherche linéaire est globalement convergente est erronée. Par conséquent, la seule globalisation connue de la méthode de Newton semi-lisse convergente est celle de Han, Pang et Rangaraj et elle peut avoir à résoudre des LCPs.

De façon similaire à la méthode de Harker et Pang, la stratégie employée par Bai et Dong est de commencer avec un point différentiable et d'éviter les plis au fil des itérations. Afin d'éviter les problèmes mentionnés à la section 1.3.3 quant au paramètre ε, ils proposent de calculer toutes les valeurs de pas de déplacement α qui font en sorte que $x + \alpha d$ est sur un pli de Θ pour les éviter. Ensuite, un pas de déplacement est choisi de sorte que Θ ait suffisamment décru.

Il est facile de démontrer que cette méthode est globalement convergente lorsque le problème admet une solution et que la séquence de pas de déplacement engendrée par l'algorithme est uniformément positive. D'ailleurs, cette démonstration est faite par Bai et Dong et est semblable à celle de Pang pour l'algorithme de B-Newton [Pan90]. Par contre, l'hypothèse sur les pas de déplacement n'est pas toujours vérifiée comme l'affirment les auteurs [BD06, Théorème 3.1]. L'exemple suivant montre qu'il est possible que l'algorithme engendre un pas de déplacement arbitrairement petit.

Exemple 1 (Pas de déplacement non uniformément positifs)**.** Considérons le problème de complémentarité linéaire de Murty [Mur78] en deux dimensions qui est défini par

$$M = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \qquad q = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

avec comme point de départ $x_t = (\frac{1}{2} - t, -\frac{1}{2})^\mathsf{T}$ avec $t > 0$.

On remarque que M est une **P**-matrice. Puisque $Mx_t + q = (\frac{1}{2} - t, -\frac{1}{2} - 2t)^\mathsf{T}$, les ensembles d'indices sont $A_0(x_t) = \varnothing$, $I_0(x_t) = \{1, 2\}$, et $E(x_t) = \varnothing$ ainsi F est différentiable en x_t et la direction de Newton-min d^{NM} est unique. Il s'ensuit que $F(x_t) = Mx_t + q$, donc d^{NM} est tel que $Mx_t + q + Md = 0$. On trouve aisément que

$$d_t^{\text{NM}} = \begin{bmatrix} \frac{1}{2} + t \\ -\frac{1}{2} \end{bmatrix}.$$

L'application F a comme unique pli la droite verticale $\{\frac{1}{2}\} \times \mathbb{R}$. Par ailleurs, la fonction $\alpha \in [0, 1] \mapsto \Theta(x_t + \alpha d_t^{\text{NM}})$ est quadratique décroissante sur $[0, \alpha_t]$ où $\alpha_t := 2t/(1 + 2t)$ est le pas de déplacement qui amène $x_t + \alpha_t d^{\text{NM}}$ sur le pli de F. Tandis que sur le segment $[\alpha_t, \infty)$ elle continue de décroître jusqu'à atteindre son minimum en

$$\alpha_t^{\min} := \frac{3t^2 + 2t}{3t^2 + 2t + \frac{1}{2}} > \alpha_t \,^1$$

et augmente de manière quadratique par la suite. On remarquera que la croissance du second segment est beaucoup plus importante que la décroissance du premier, tellement que cela entraîne qu'il est possible de choisir x_t assez proche du pli de sorte que le pas de déplacement est inférieur à toute valeur $\varepsilon > 0$. D'ailleurs $\alpha_t^{\min} \to 0$ lorsque $t \downarrow 0$ ce qui laisse présager que des pas infiniment petits peuvent être engendrés. Pour voir que $x + \alpha_t d^{\text{NM}}$ est bien un pli de Θ, calculons $\check{x}_t := x_t + \alpha_t d_t^{\text{NM}} = (\frac{1}{2}, (-2t - \frac{1}{2})/(1 + 2t))^\mathsf{T}$ et

$$F(\check{x}_t) = \begin{bmatrix} -\frac{1}{2} \\ \frac{-2t-\frac{1}{2}}{1+2t} \end{bmatrix}, \quad F'(\check{x}_t; d^{\text{NM}}) = \begin{bmatrix} \frac{1}{2} + t \\ -\frac{1}{2} \end{bmatrix}, \quad F'(\check{x}_t; -d^{\text{NM}}) = \begin{bmatrix} -\frac{1}{2} - t \\ -\frac{1}{2} - 2t \end{bmatrix}.$$

1. La valeur $\alpha_t^{\min}$ est obtenue en faisant l'interpolation quadratique à partir des points $\check{x}_t = x_t + \alpha_t d_t^{\text{NM}}$ et $x_t + d^{\text{NM}}$.

Les ensembles d'indices sont $A_0(\check{x}_t) = \varnothing$, $I_0(\check{x}_t) = \{1\}$, et $E(\check{x}_t) = \{2\}$. Donc les dérivées directionnelles sont données par :

$$\Theta'(\check{x}_t; -d_t^{\text{NM}}) = F(\check{x}_t)^\mathsf{T} F'(\check{x}_t; -d^{\text{NM}}) = \frac{5t^2 + 3t + \frac{1}{2}}{1 + 2t} > \frac{1}{2}$$
$$\Theta'(\check{x}_t; d_t^{\text{NM}}) = F(\check{x}_t)^\mathsf{T} F'(\check{x}_t; d^{\text{NM}}) = \frac{-t^2}{1 + 2t} < 0.$$

En effet, il y a un saut dans les dérivées directionnelles ce qui confirme que $\check{x}_t$ est bien un pli de F. De plus, étant donné la valeur de $\Theta'(\check{x}_t; d_t^{\text{NM}})$, on remarque que la décroissance devient de plus en plus faible immédiatement après le pli et tend à être nulle à mesure que $t \downarrow 0$.

Sachant que $\Theta(x_t + \alpha d^{\text{NM}})$ est décroissant jusqu'à $\Theta(x_t + \alpha_t^{\min} d^{\text{NM}})$ et crossant par la suite, on cherche $\bar{\alpha}_t > \alpha_t^{\min}$ tel que $\Theta(x_t) = \Theta(x_t + \bar{\alpha}_t d^{\text{NM}})$. Notons qu'après le pli, les indices sont répartis de la manière suivante : $A_0(x_t) = \{2\}$, $I_0(x_t) = \{1\}$, et $E(x_t) = \varnothing$. D'où

$$\begin{aligned} F(x_t + \bar{\alpha}_t d^{\text{NM}}) &= \begin{bmatrix} (M(x_t + \bar{\alpha}_t d^{\text{NM}}) + q)_1 \\ (x_t + \bar{\alpha}_t d^{\text{NM}})_2 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2} - t + \bar{\alpha}_t(\frac{1}{2} + t) \\ -\frac{1}{2} - \frac{1}{2}\bar{\alpha}_t \end{bmatrix}. \end{aligned}$$

Ainsi,

$$\begin{aligned} 2\Theta(x_t + \bar{\alpha}_t d^{\text{NM}}) &= F(x_t + \bar{\alpha}_t d^{\text{NM}})^\mathsf{T} F(x_t + \bar{\alpha}_t d^{\text{NM}}) \\ &= (-\tfrac{1}{2} - t + \bar{\alpha}_t(\tfrac{1}{2} + t))^2 + (-\tfrac{1}{2} - \tfrac{1}{2}\bar{\alpha}_t)^2 \\ &= \tfrac{1}{2} + t + t^2 - \bar{\alpha}_t(\tfrac{1}{2} + t) - 2t\bar{\alpha}_t(\tfrac{1}{2} + t) + \bar{\alpha}_t^2(\tfrac{1}{2} + t)^2 + \tfrac{1}{4}\bar{\alpha}_t^2 \\ &= (\tfrac{1}{2} + t + t^2)\bar{\alpha}_t^2 + (-2t - 2t^2)\bar{\alpha}_t + (\tfrac{1}{2} + t + t^2). \end{aligned}$$

De plus, initialement la fonction de mérite est $\Theta(x_t) = (Mx + q)^\mathsf{T}(Mx + q) = 5t^2 + 3t + \frac{1}{2}$.

D'où $\Theta(x_t) = \Theta(x_t + \bar{\alpha}_t d^{\text{NM}})$ si et seulement si

$$\begin{aligned}
&5t^2 + 3t + \tfrac{1}{2} = (\tfrac{1}{2} + t + t^2)\bar{\alpha}_t^2 + (-2t - 2t^2)\bar{\alpha}_t + (\tfrac{1}{2} + t + t^2) \\
&\Leftrightarrow 0 = (\tfrac{1}{2} + t + t^2)\bar{\alpha}_t^2 + (-2t - 2t^2)\bar{\alpha}_t + (-4t^2 - 2t) \\
&\Leftrightarrow \bar{\alpha}_t = \frac{2t(t+1) + \sqrt{(-2t^2 - 2t)^2 - 4(t^2 + t + \frac{1}{2})(-4t^2 - 2t)}}{2t^2 + 2t + 1} \\
&\qquad = \frac{2t(t+1) + 2\sqrt{t(5t^3 + 8t^2 + 5t + 1)}}{2t^2 + 2t + 1}.
\end{aligned}$$

Puisque $\bar{\alpha}_t \to 0$ lorsque $t \downarrow 0$, on conclut qu'un pas de déplacement qui assure la décroissance de Θ peut être arbitrairement petit. □

Bibliographie

[BD06] Z.-Z. Bai and J.-L. Dong. A Modified Damped Newton Method for Linear Complementarity Problem. *Numerical Algorithms*, 42 :207–228, 2006.

[BE18] Q.M. Bui and H.C. Elman. Semi-smooth Newton Methods for Nonlinear Complementarity Formulation of Compositional Two-phase Flow in Porous Media. Technical report, 2018.

[BG12] I. Ben Gharbia and J.Ch. Gilbert. A Simple Example of Semismooth Newton Direction that is not a Descent Direction of the Least-square Merit Function, 2012. Working notes.

[BG13] I. Ben Gharbia and J.Ch. Gilbert. An Algorithmic Characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3) :904–916, 2013.

[BG18] I. Ben Gharbia and J.Ch. Gilbert. An Algorithmic Characterization of P-matricity II : Corrections, Refinements, and Validation. *SIAM Journal on Matrix Analysis and Applications*, 2018.

[BGS13] A.P. Bourgeat, S. Granet, and F. Smaï. *Compositional Two-phase Flow in Saturated-Unsaturated Porous Media : Benchmarks for Phase Appearance/Disappearance*, pages 81–106. De Gruyter, 2013.

[BJ14] I. Ben Gharbia and J. Jaffré. Gas Phase Appearance and Disappearance as a Problem with Complementarity Constraints. *Mathematics and Computers in Simulation*, 99 :28–36, 2014.

[BKKK11] H. Buchholzer, C. Kanzow, P. Knabner, and S. Kräutle. The Semismooth Newton Method for the Solution of Reactive Transport Problems Including Mineral Precipitation-Dissolution Reactions. *Computational Optimization and Applications*, 50(2) :193–221, 2011.

[CCK00] B. Chen, X. Chen, and C. Kanzow. A Penlaized Fischer-Burmeister NCP-function. *Mathematical Programming*, 88(1) :211–216, 2000.

[Cla83] F.H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York, 1983.

[CPS92] R.W. Cottle, J.-S. Pang, and R.E. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

[DF95] S.P. Dirkse and M.C. Ferris. The Path Solver : a Nommonotone Stabilization Scheme for Mixed Complementarity Problems. *Optimization Methods and Software*, 5(2) :123–156, 1995.

[DFG18] J.-P. Dussault, M. Frappier, and J. C. Gilbert. Globally Convergent Modification of the Semismooth Newton Method Applied to Complementarity Problems. 2018. Cet article soumis sera soumis à SIAM journal of Optimization.

[Erl13] K. Erleben. Numerical Methods for Linear Complementarity Problems in Physics-based Animation, Febuary 2013. Siggraph course notes.

[Fat79] Y. Fathi. Computational Complexity of LCPs Associated with Positive Definite Symmetric Matrices. *Mathematical Programming*, 17(2) :335–344, 1979.

[Fis92] A. Fischer. A Special Newton-type Optimization Method. *Optimization*, 24(3-4) :269–284, 1992.

[FP62] M. Fielder and V. Pták. On Matrices with Nonpositive off-diagonal Elements and Principal Minors. *Czechoslovak Mathematics Journal*, 12(3) :382–400, 1962.

[FP97] M. C. Ferris and J. S. Pang. Engineering and Economic Applications of Complementarity Problems. *SIAM Review*, 39(4) :669–713, 1997.

[FP03] F. Facchinei and J. S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume 2. Springer-Verlag New York, 2003.

[HP90] P.T. Harker and J.-S. Pang. A Damped-Newton Method for the Linear Complementarity Problem. In *Computational Solution of Nonlinear Systems of Equations*, number 26 in Lecture in Applied Mathematics. AMS, Providence, RI, 1990.

[HPR92] S.-P Han, J.-S. Pang, and N. Rangaraj. Convergent Newton Methods for Nonsmooth Equation. *Mathematics of Operations Research*, 17(3) :586–607, 1992.

[Kar39] W. Karush. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master's thesis, Département de mathématiques, Université de Chicago, 1939.

[KT51] H. W. Kuhn and A. W. Tucker. Nonlinear Programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, 1951.

[Lem65] C. E. Lemke. Bimatrix Equilibrium Points and Mathematical Programming. *Management Science*, 11(3) :681–689, 1965.

[LFK00] T. De Luca, F. Facchinei, and C. Kanzow. A Theoretical and Numerical Comparison of Some Semismooth Algorithms for Complementarity Problems. *Computational Optimization and Applications*, 16(2) :173–205, 2000.

[LJ64] C. E. Lemke and J. T. Howson Jr. Equilibrium Points of Bimatrix Games. *Management Science*, 12(2) :413–423, 1964.

[Mur78] K.G. Murty. Computational Complexity of Complementarity Pivot Methods. *Mathematical Programming Study*, 7 :61–73, 1978.

[Pan90] J.-S. Pang. Newton's Method for B-differentiable Equations. *Mathematics of Operations Research*, 15(2) :311–341, 1990.

[Pan91] J.-S. Pang. A b-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems. *Mathematical Programming*, 51 :101–131, 1991.

[War52] J. G. Wardrop. Some Theoretical Aspects of Road Traffic Research. *Proceeding of the Institute of Civil Engineers, Part II*, pages 325–378, 1952.

[Wri94] S. Wright. An Infeasible Interior-Point Algorithm for Linear Complementarity Problem. *Mathematical Programming*, 67(1-3) :29–51, 1994.

[Zha94] Y. Zhang. On the Convergence of a Class of Infeasible Interior-Point Methods for the Horizontal Linear Complementarity Problem. *SIAM Journal of Optimization*, 4(1) :208–227, 1994.